LifeStyle Store

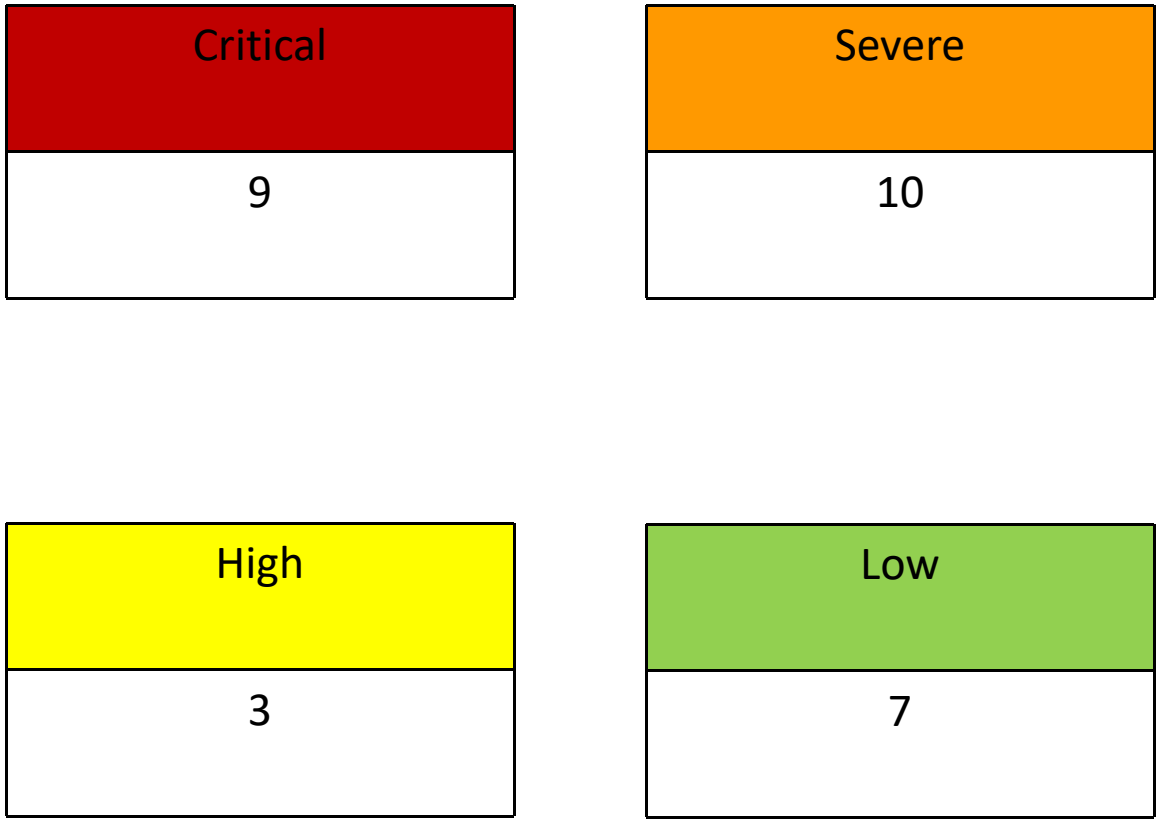
E-Commerce Plateform

Detailed Developer Report

Security status –Extremely Vulnerable

- Hacker can steal all records in LifeStyle databases (SQLi)
- Hacker can take control of complete server including View, Add, Edit, Delete files and folders (Shell Upload)
- Hacker can change source code of application to host malware, phishing pages or even explicit content (Shell Upload)
- Hacker can get account details of another customer like by changing the number in edit profile link (IDOR)
- Hacker can send multiple requats (Rate Limiting Flaw)
- Hacker can add/remove items in cart (CSRF)
- use of http intead of https

Vulnerability statistics



Vulnerabilities

Sr.no.	Severity	Vulnerabilities	count
1	Critical	SQL injections	2
2	Critical	Remote file inclusion	1
3	Critical	Admin panel access	1
4	Critical	Insecure /Arbitrary File Uploads	1
5	Critical	Unauthorised Access To Seller account access	1
6	Critical	Components with known vulnerability	3
7	Severe	Crypto Configuration Flaws	1
8	Severe	C.S.R.F	2
9	Severe	Coupon code brute force	1

Vulnerabilities

Sr.no.	Severity	Vulnerabilities	count
11	Severe	Insecure direct object references	3
12	Severe	Open redirection	1
13	Severe	Cross site scripting	2
14	High	Client side filter bypass	1
15	High	Directory listing	1
16	High	Personnel identifiable information	1
17	Low	Default and debug pages	5
18	Low	Default error display	2

1. S.Q.L. Injections

SQL

The bellow mentioned url is vulnerable to sql injections

- Affacted url
  http://13.235.132.114/products.php?cat=(here)

- Affacted parameters

1. cat

- Payload
  cat=3’

- Affacted url
  http://13.235.132.114/search/search.php?q=(here)

- Affacted parameters

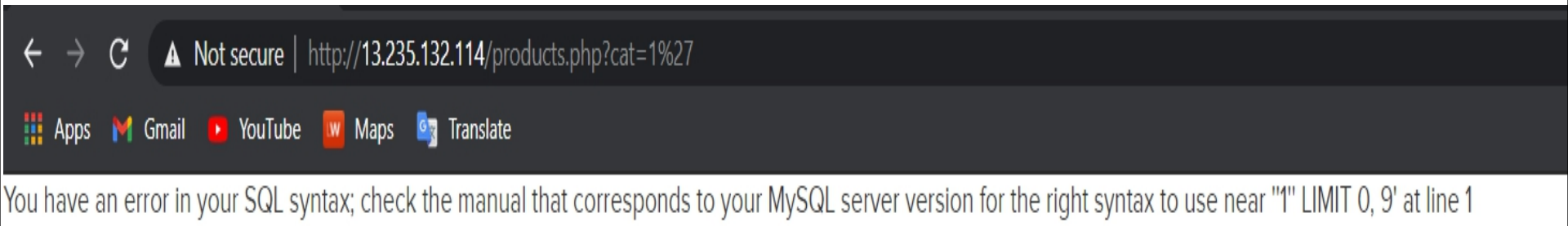
1. q

- Payload
  q=woodland’

Observations

At home page click on any one category.

Notice the get category of cat and add ‘ and then observe the error.



Observations

- In this URL : http://13.235.132.114/products.php?cat=3

- We apply single quote in house parameter: **products.php?cat=3' and we get complete MySQL error: (img 1)**

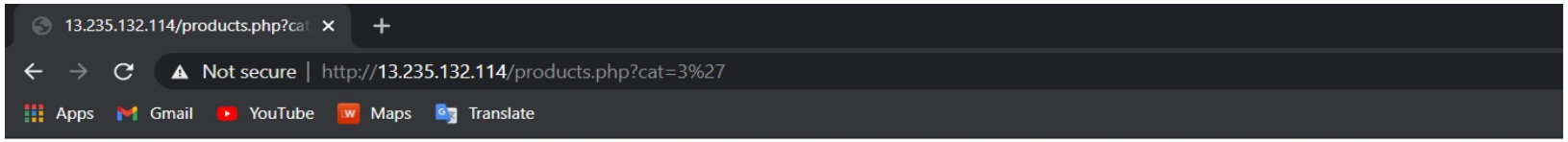
- We then put --+ : **products.php?cat=3'--+ and the error is removed confirming SQL injection:(img 2)**

- When we write http://13.235.132.114/products.php?cat=3' --+ the error would be removed

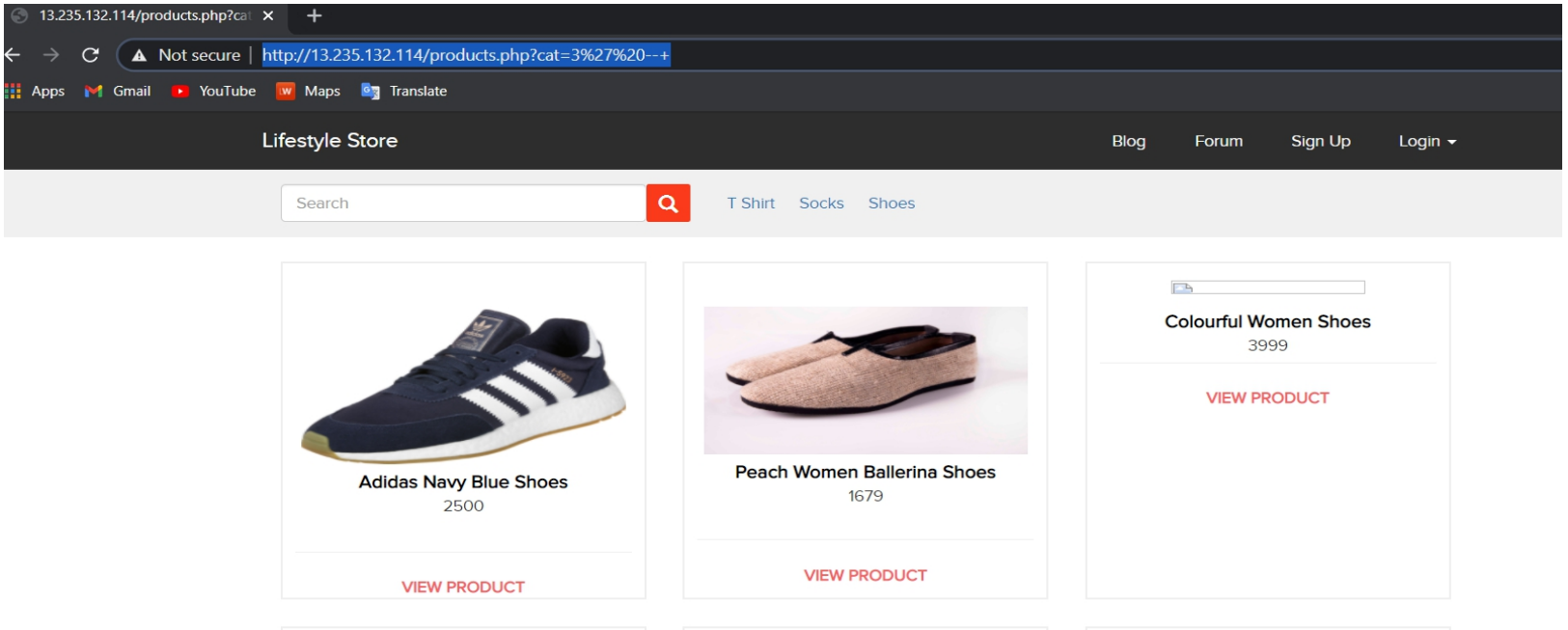
That confirms the sql injection

Observations

- img1:



- img2:

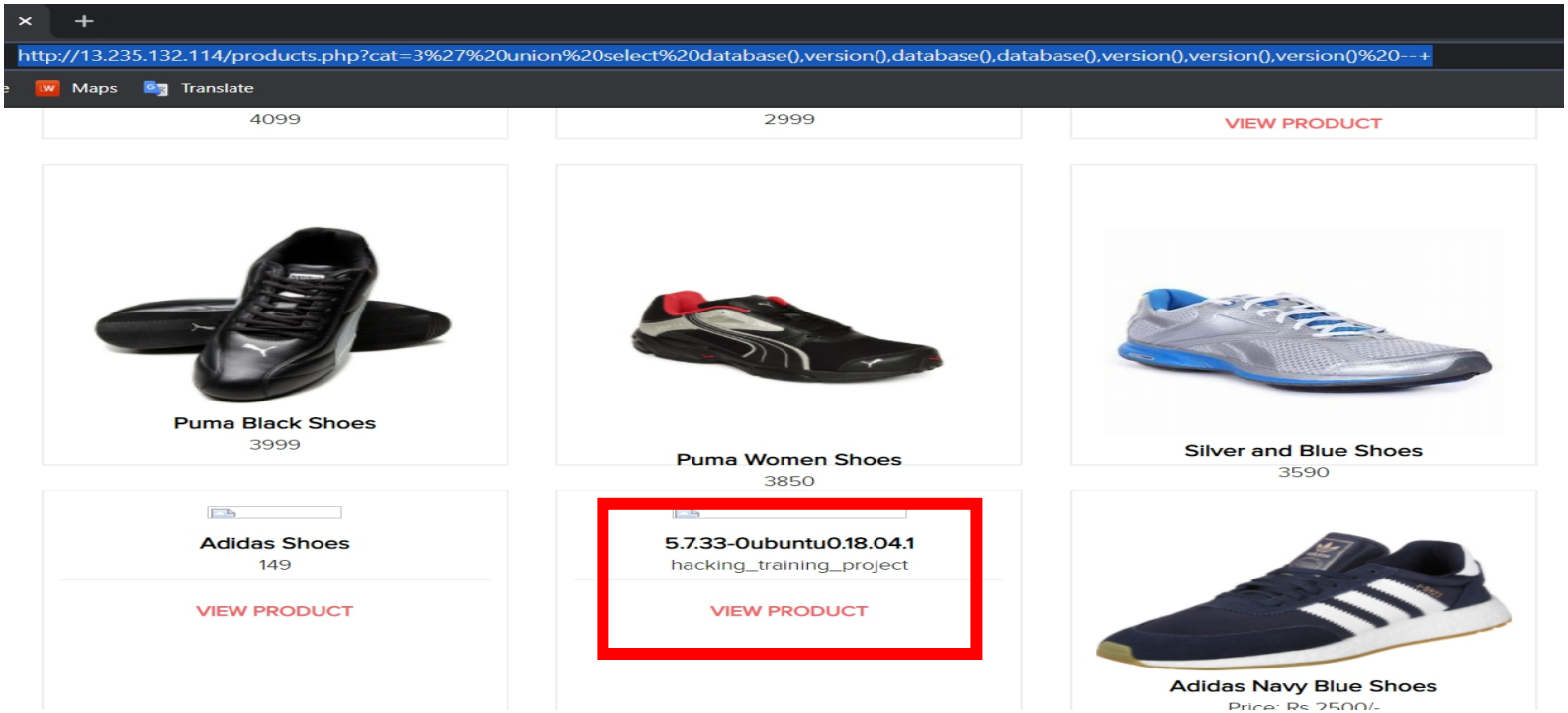


Proof of concept

Attacker can execute SQL commands as shown below. Here we have used the payload below to extract the database name and

MySQL version information:

http://13.235.132.114/products.php?cat=3%27%20union%20select%20database(),version(),database(),database(),version(),version(),version()%20--+



Proof of concept

No of databases: 2

- information_schema
- hacking_training_project

No of tables in SQL_Injection_V3: 10

- brands
- cart_items
- categories
- customers
- order_items
- orders
- product_reviews
- products
- sellers
- users

user_name	type	password	phone_number
admin	admin	$2y$10$Phrdr2F1sC912mG6jY5af.QbdJ7O6yasyHc/CZiNEchBPsWJiWuK2	8521479630
Dona1234	customer	$2y$10$PM.7nBSP5FMa1dXiM/S3s./p5xR6GTKvjry7ysJtxOkBqOJURAHsO	9489625136
Pluto98	customer	$2y$10$ba4bpp3nqfFRPB9.w.s4KeU36ecbRemyM6bj65FI/Q1Et0Qv1X9QK	8912345670
chandan	seller	$2y$10$4cZBEIrgthXdvT1hwUlivuFELe03rR.GIcdp03NjrlS0VeiOKLVDa	7854126395
Popeye786	customer	$2y$10$Fkv1RfwYTioW0w2CaZtAQuXVnhGAUjt/If/yTqkNPC5zTrsVm7EeC	9745612300
Radhika	seller	$2y$10$RYxNhOyV/G4g7OtFwpqYaexvHi8rF6XXui8kT1WtrfqhTutCA8JC.	9512300052
Nandan	seller	$2y$10$G.cRNLMEiG79ZFXElHg.R.o95334U0xmZu4.9MqzR5614ucwnk59K	7845129630
MurthyAdapa	customer	$2y$10$mzQGzD4sDSj2EunpCioe4eK18c1Abs0T2P1a1P6ev1DPR.11UubDG	8365738264
john	customer	$2y$10$GhDB8h1X6XjPMY12GZ1vDO7Y3en97u1/.oXTZLmYqB6F18FBgecvG	6598325015
bob	customer	$2y$10$kiUikn3HPFbuyTtK75lLNurxzqC0LX3eMGy0/Ux16JOoG37dCGKLq	8576308560
jack	customer	$2y$10$z/nyNlkRJ76m9ItMZ4N5lOeRxy6Gkqi9N/UBcJu5ZeO7eM7N4pTHu	9848478231
bulla	customer	$2y$10$HT5oiRMetqaZ7xGZPE9s2.Mk1yF4PnYDJHCWbm2w/xuKpjEEI/zjG	7645835473
hunter	customer	$2y$10$pB3U9iFxwBgSbl2AkBpiEeIBdhiYfWy9y.xV23q12gGbMCyn7N3g2	9788777777
asd	customer	$2y$10$At5pFZnRWpjCD/yNnJWDL.L3Cc4Cv0W8Q/WEHmWzBFqVIkBQFpCF2	9876543210
acdc	customer	$2y$10$J50B78.gpucuLTwpHwbcPedYcain.Yi.tsTLyQtKl7FzdSpmIRRbi	9999999999
FindMe	customer	$2y$10$ieLZsBhtXY0N92Wyo3o5y.BQJO4zd7tpcF18XV61F/FhyBT6.zfNa	9999999999

Business impact- Extremely High

Using this vulnerability, attacker can execute arbitrary SQL commands on Lifestyle store server and gain complete access to internal databases along with all customer data inside it.

Below is the screenshot of some information extracted from users table which shows user credentials being leaked .Since the passwords are hashed ,the risk is comparatively low .

Attacker can use this information to attack the users and login to admin panels and gain complete admin level access to the website which could lead to complete compromise of the server and all other servers connected to it.

http://13.235.132.114/products.php?cat=1%27%20union%20select%201,user_name,3,password,5,6,7%20from%20users%20--+

Recommendations

Take the following precautions to avoid exploitation of SQL injections:

•Whitelist User Input: Whitelist all user input for expected data only. For example if you are expecting a flower
name, limit it to alphabets only upto 20 characters in length. If you are expecting some ID, restrict it to numbers
only
•Prepared Statements: Use SQL prepared statements available in all web development languages and frameworks
to avoid attacker being able to modify SQL query
•Character encoding: If you are taking input that requires you to accept special characters, encode it. Example.
Convert all ‘ to \’ , “ to \”, \ to \\. It is also suggested to follow a standard encoding for all special characters such
has HTML encoding, URL encoding etc
•Do not store passwords in plain text. Convert them to hashes using SHA1 SHA256 Blowfish etc
•Do not run Database Service as admin/root user
•Disable/remove default accounts, passwords and databases
•Assign each Database user only the required permissions and not all permissions.

References

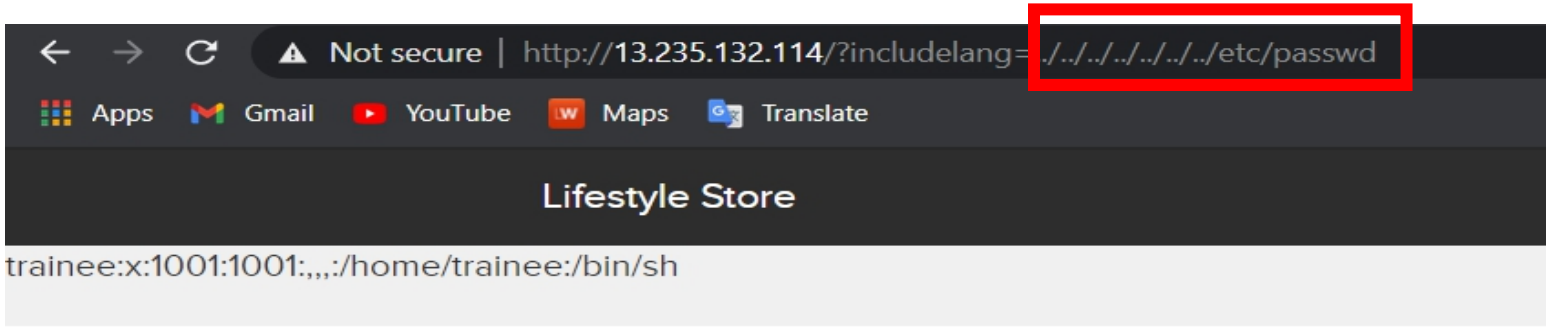
- *https://www.owasp.org/index.php/SQL_Injection*
- *https://en.wikipedia.org/wiki/SQL_injection*

2.Remote File inclusion

RFI	Below mentioned url is vulnerable to RFI • Affected url http://13.235.132.114/?includelang=(here) Payload ../../../../../../../etc/passwd

Observations

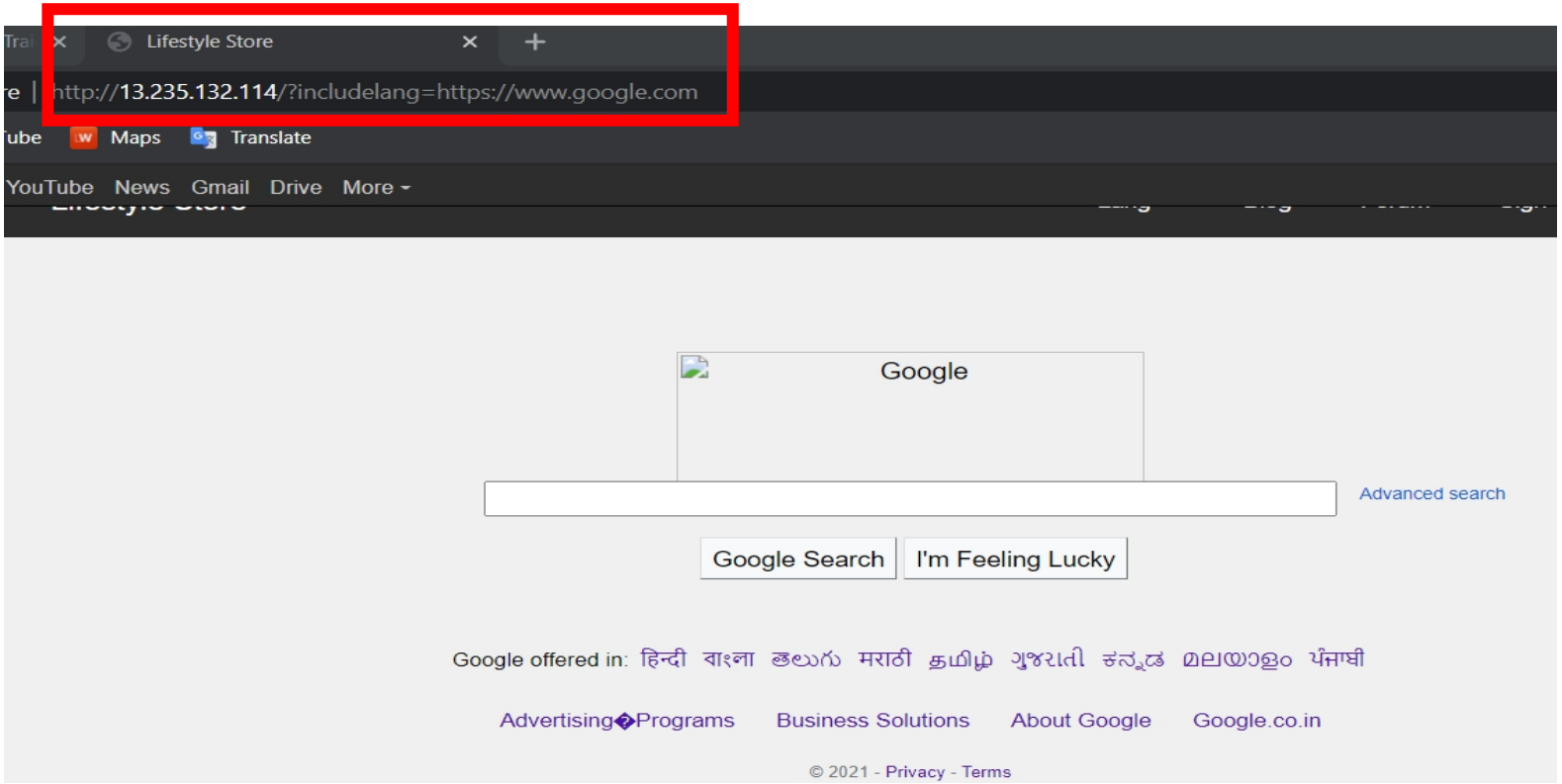
When you click on change language you get a ‘get’ parameter of includelang which is vulnerable for file inclusion.



Here we have tested Local file inclusion to execute a file which gives us user name

POC-attacker can upload shells

- **Attacker can exploit** the referencing function in an application to upload malware (e.g., backdoor shells) from a remote URL located within a different domain.



Business impact- Extremely high

- Any attacker can have the root access of of your website
- He can execute commands
- Through the website he can have access of the server and can infect other websites hosted on that server
- He can even deface your websites

Recommendations

- To safely parse user-supplied filenames it's much better to **maintain a whitelist of acceptable filenames** and use a corresponding identifier (not the actual name) to access the file. Any request containing an invalid identifier can then simply be rejected. This is the approach that OWASP recommends.

References

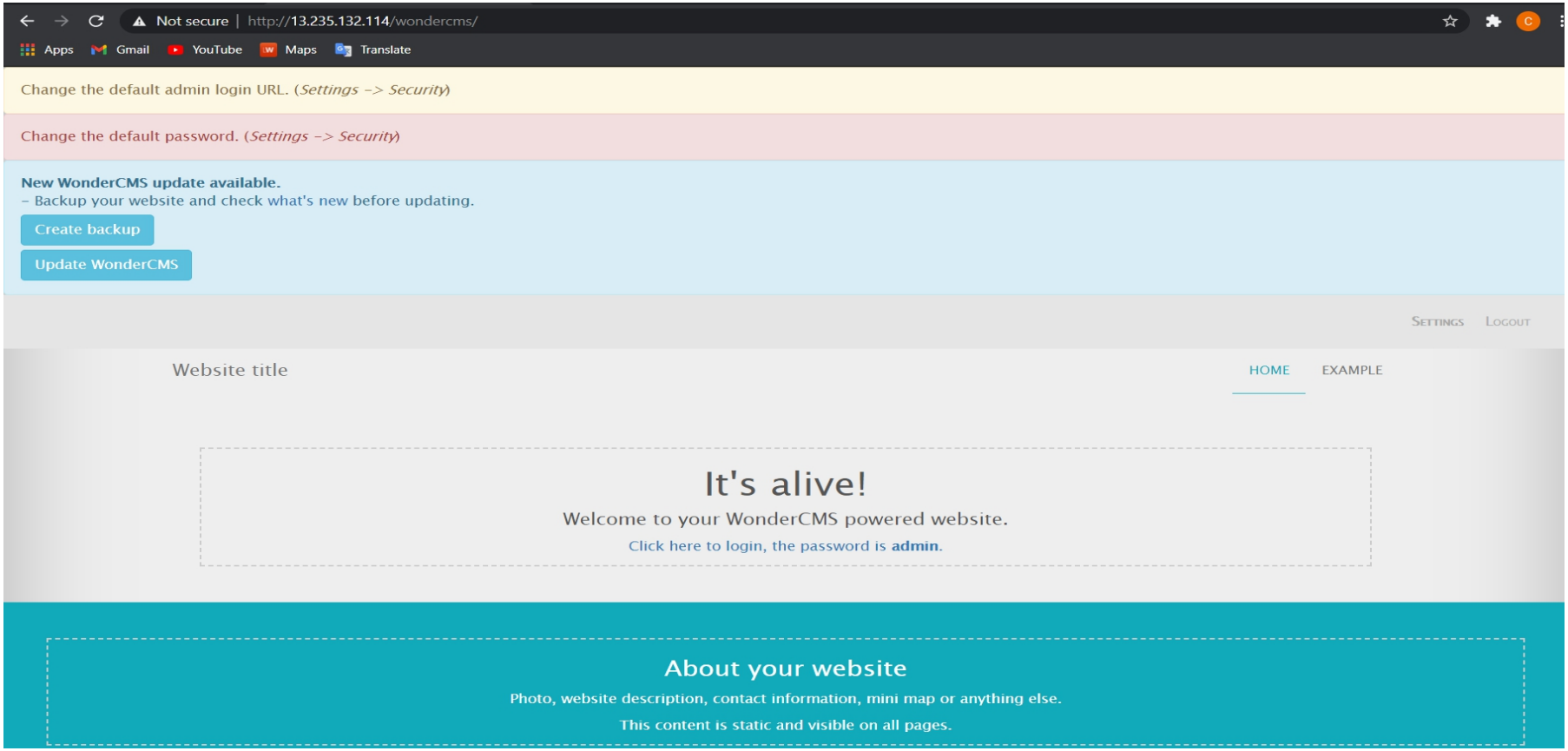
- https://www.pivotpointsecurity.com/blog/file-inclusion-vulnerabilities/
- https://www.netsparker.com/blog/web-security/local-file-inclusion-vulnerability/
- https://en.wikipedia.org/wiki/File_inclusion_vulnerability

3.Admin panel access

Admin panel access	Below mentioned URL is vulnerable to Arbitrary File Upload and making other admin level changes. • Affected url http://13.235.132.114/wondercms/loginURL

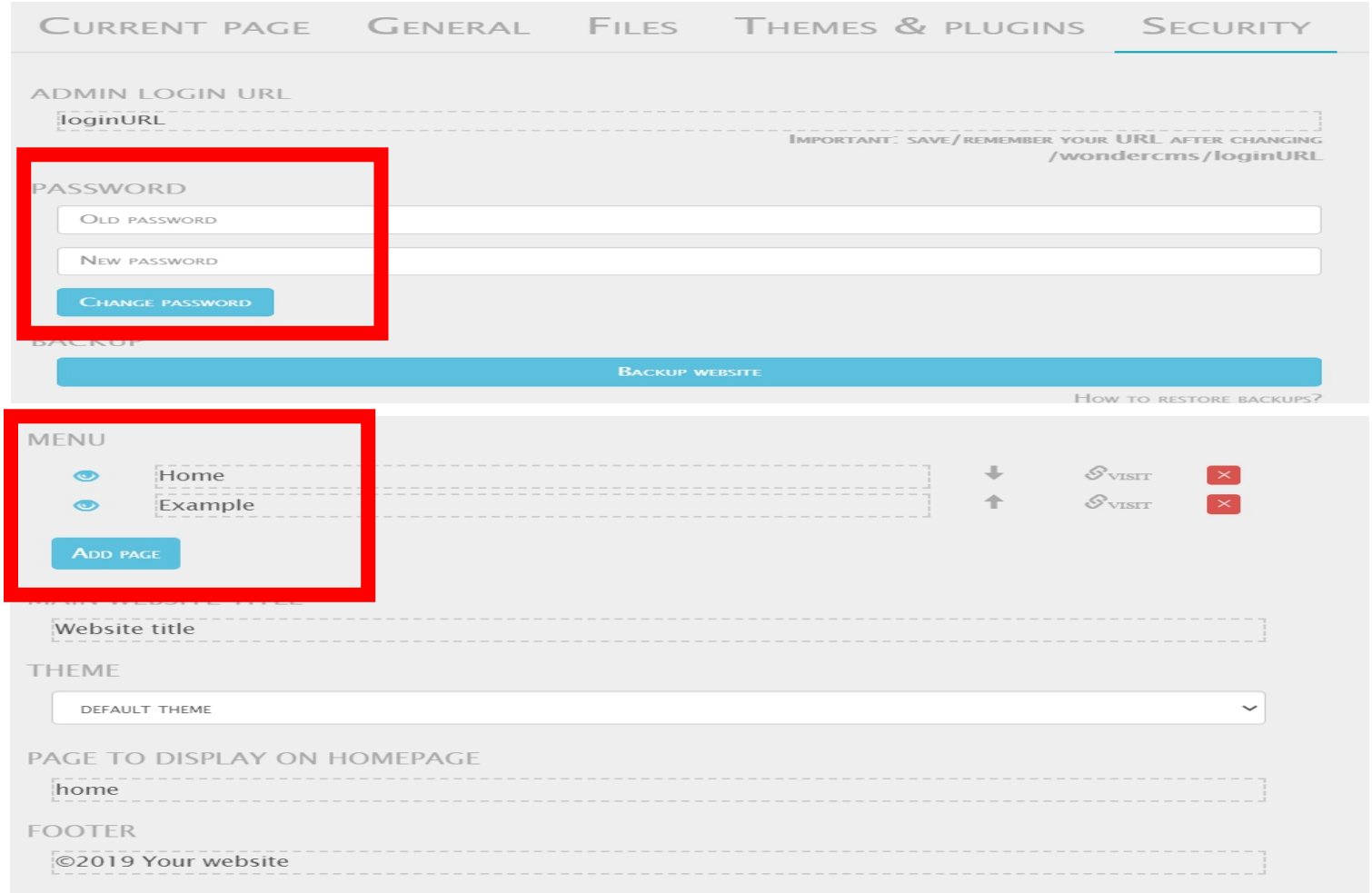
Observation

- When we navigate to http://13.235.132.114/wondercms/ url ,we get the password on the page and login as : admin
- in the url http://13.235.132.114/wondercms/loginURL .

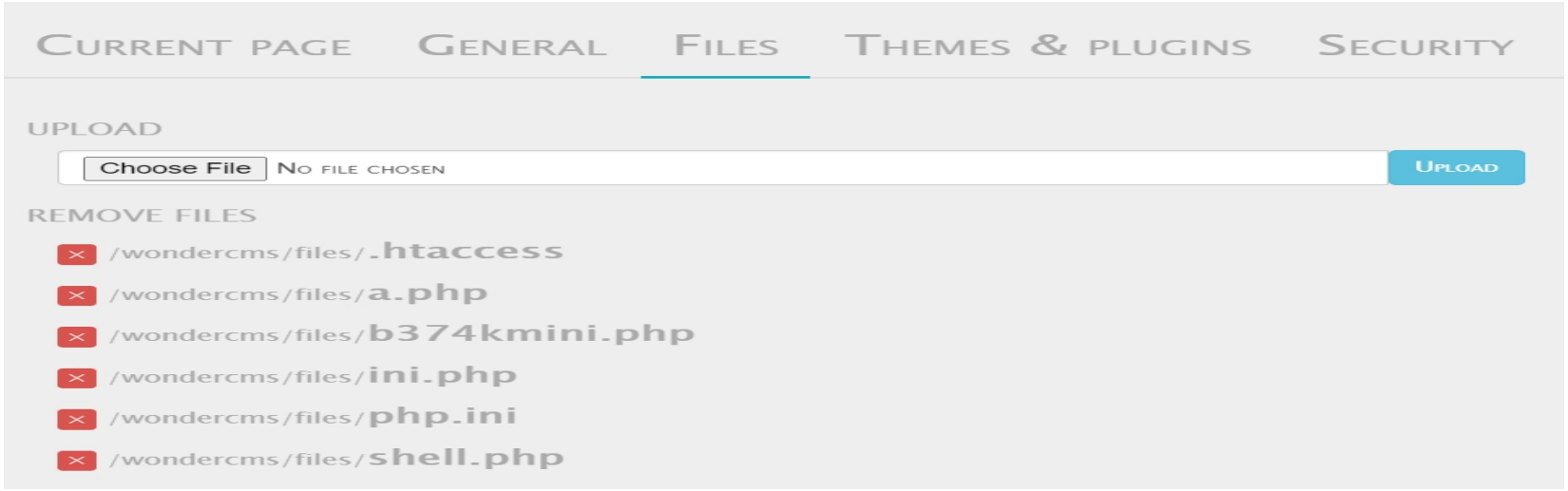


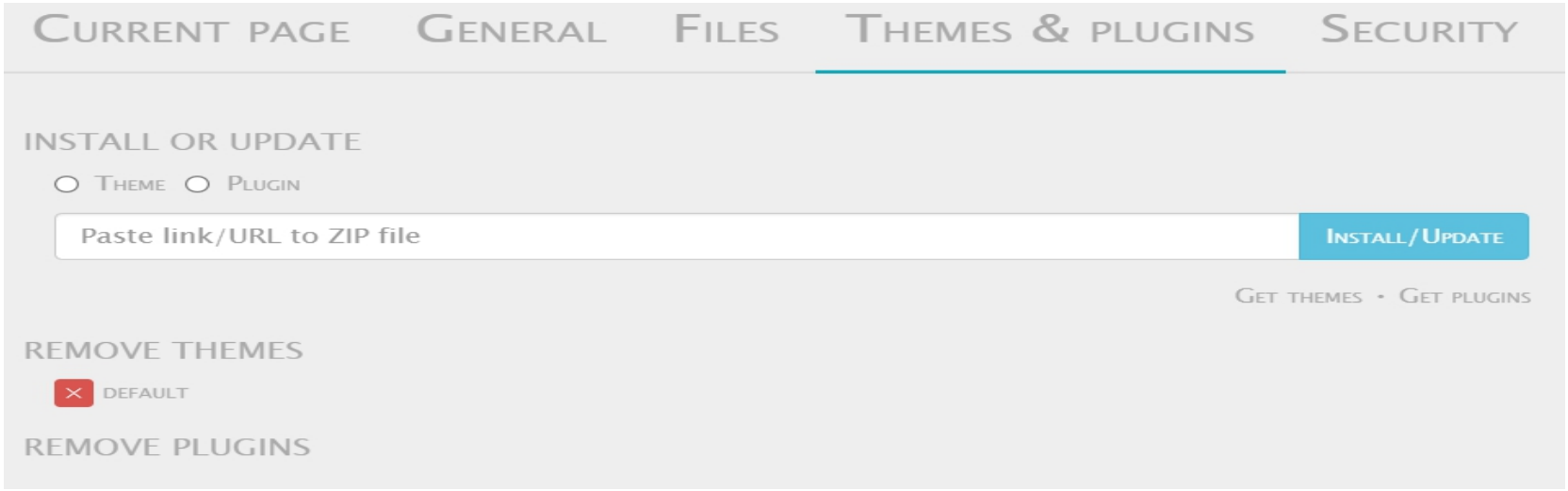
Proof of Concept (PoC)

- Hacker can change the admin login password making the actual admin unable to login the next time . Hacker can also add and delete pages.
- Here is the admin dashboard:-



Proof of Concept (PoC)





Business impact-extremely high

- Using this vulnerability ,the attacker can get complete access to the blog of the website.
- The attacker can change the password or even change the url of the admin panel and restrict the admin to access it.
- Even pages can be created and deleted along with editing.
- Files can be added (without verification) and hence can be dangerous to the entire website,as the control of the entire website can be taken.

Recommendations

- The default password should be changed and a strong password must be setup.
- The admin url must also be such that its not accessible to normal users.
- Password changing option must be done with 2 to 3 step verification.
- Password must be at least 8 characters long containing numbers,alphanumerics,etc.
- All the default accounts should be removed.
- Password should not be reused.

Refernces

- https://www.owasp.org/index.php/Testing_for_weak_password_change_or_reset_functionalities_(OTG_x0002_AUTHN-009)
- https://www.owasp.org/index.php/Default_Passwords
- https://www.us-cert.gov/ncas/alerts/TA13-175A

4.Insecure/Arbitrary file uploads

Insecure file uploads	The attacker can upload insecure shells and files and gain access over the entire database and login as the admin and the vesion is known to have vulnerabilities • Affected url http://13.235.132.114/wondercms/ • Affected Parameters : File Upload (POST parameter) • Uploaded file backdoor shell

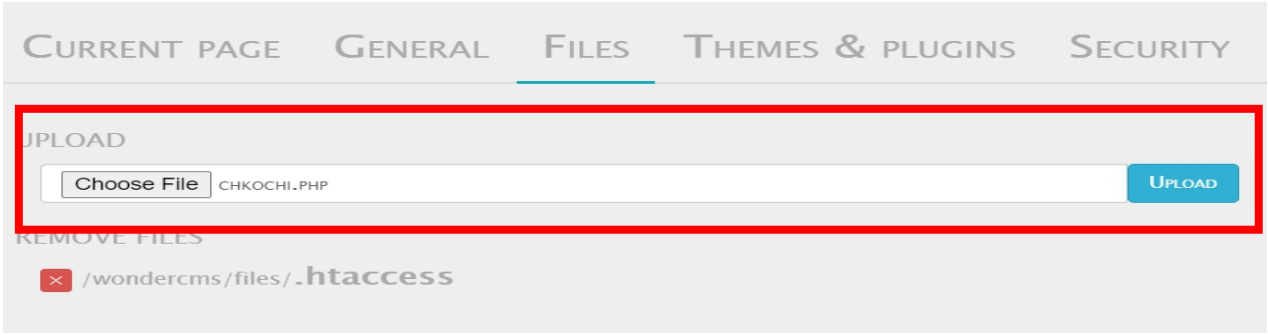
Observations

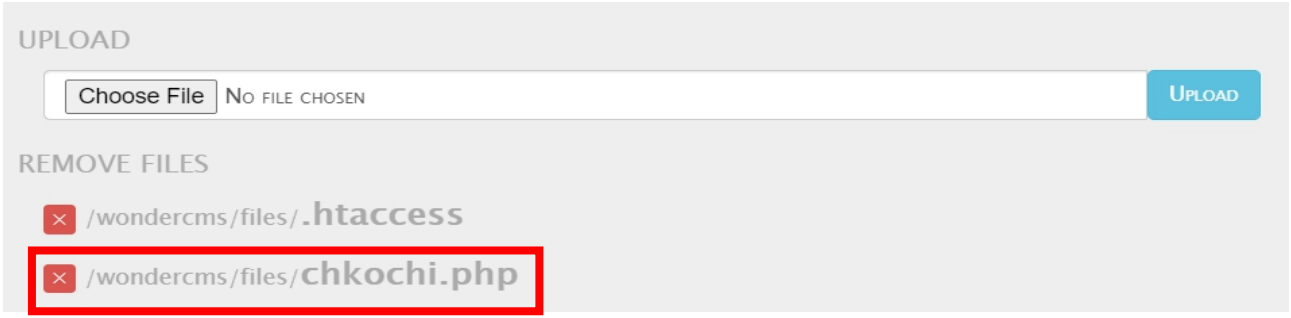
• Login using any of the user's details on the Lifestyle store and navigate to Blog tab . Now click on Login and

put the password - admin.

• Then click on Settings tab and Click on Files tab . Here hacker can upload the file like shown .

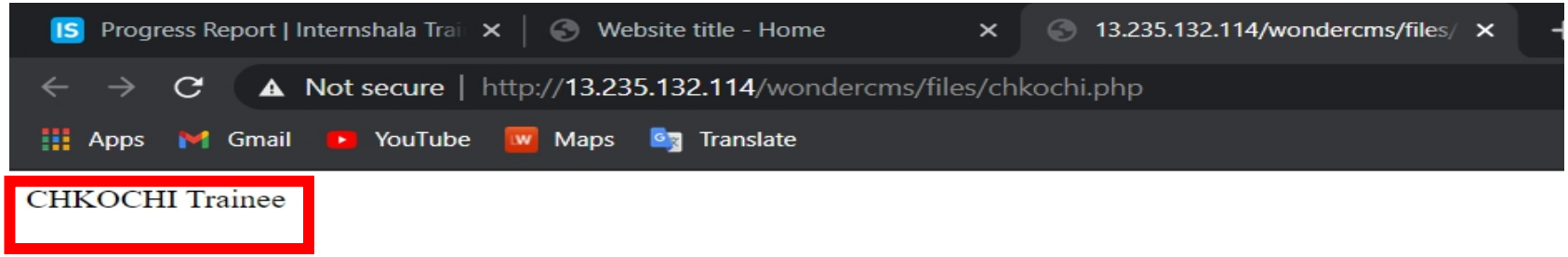
• Click on the uploaded file chkochi.php and it will be opened.





POC - Any command can be executed

- The chkochi.php I uploaded was executed successfully
- Weak password - admin.
- Arbitrary File Inclusion.
- Below is the result of the uploded file in the previous slide likewise some malicious shell can be uploaded as well.



Business impact-Extremely high

The consequences of unrestricted file upload can vary:-

- The presence of an actual malicious file can compromise the entire system leading to system takeover/ data stealing
- forwarding attacks to back-end systems
- client-side attacks, or simple defacement.
- Any backdoor file or shell can be uploaded to get access to the uploaded file on remote server and data can be exfiltrated.

Recommendations

- Never accept a filename and its extension directly without having a whitelist filter.
- The application code should be configured in such a way, that it should block uploading of malicious files extensions such as exe/ php and other extensions with a thorough server as well as client validation. CVE ID allocated: CVE-2017-14521.
- Change the Admin password to something strong and not guessable.
- Rename the files using a code, so that the attacker cannot play around with file names.
- Use static file hosting servers like CDNs and file clouds to store files instead of storing them on the application server itself

References

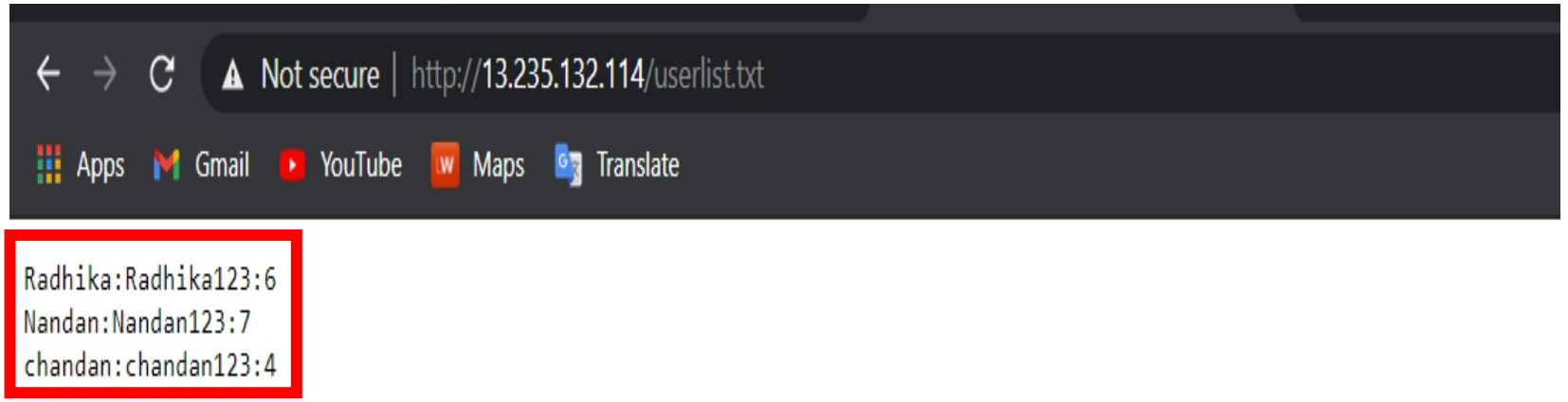
- IIS 6.0 Security Best Practices[http://technet.microsoft.com/en-us/library/cc782762(WS.10).aspx]
- Securing Sites with Web Site Permissions[http://technet.microsoft.com/en-us/library/cc756133(WS.10).aspx]
- IIS 6.0 Operations Guide[http://technet.microsoft.com/en-us/library/cc785089(WS.10).aspx]
- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- https://www.owasp.org/index.php/Unrestricted_File_Upload
- https://www.opswat.com/blog/file-upload-protection-best-practices

5.Unauthorised Access To Seller account access

Seller account access	The default page given below shows the seller accounts and passwords • Affected url http://13.235.132.114/userlist.txt http://13.235.132.114/seller/dashboard.php

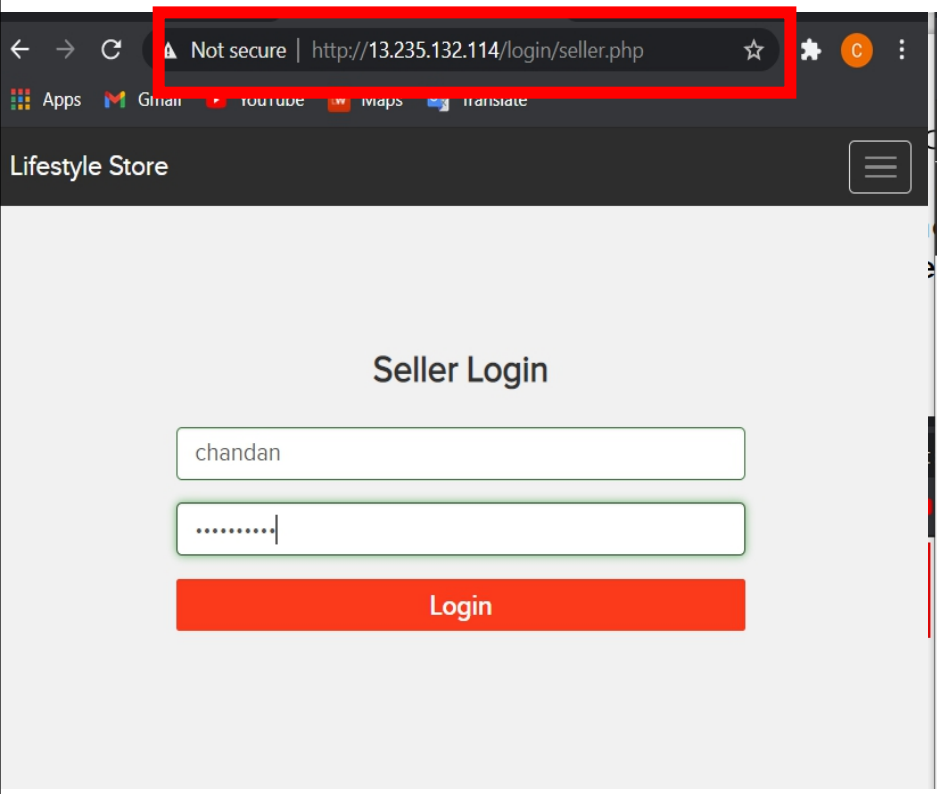
Observations

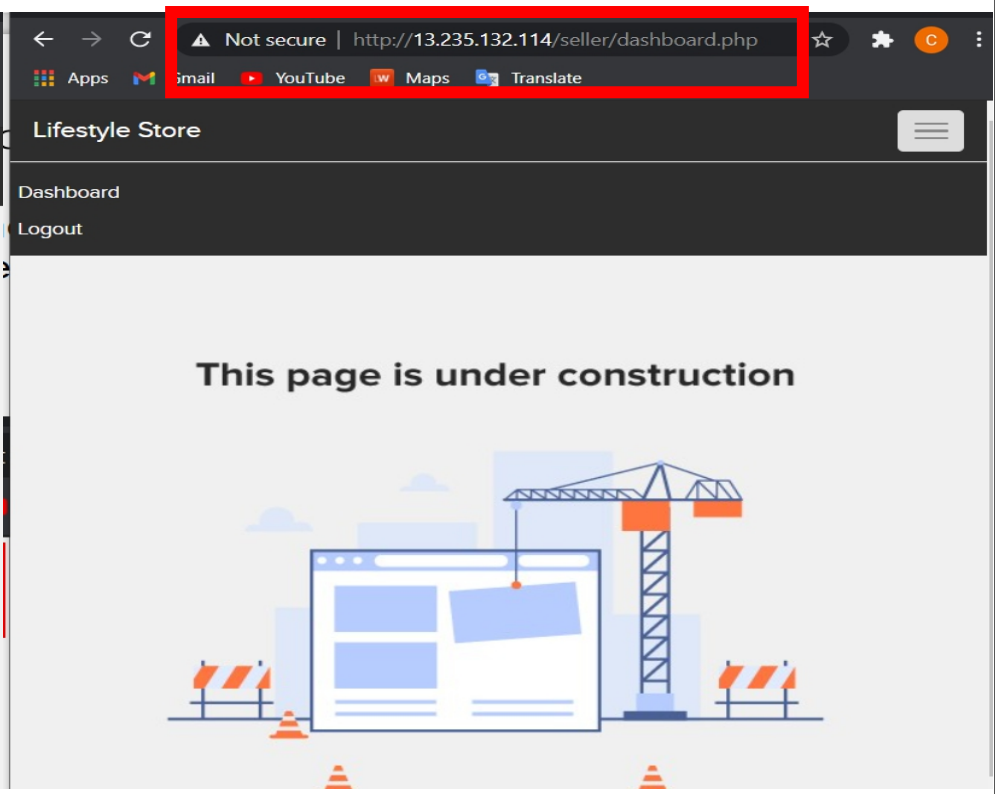
At the homepage after adding userlist.txt the following page is opened



POC-attacker has the seller dashboard acess

- On entering the credentials in the seller account login we have accessed the dashboard





Bussiness impact-Extremely high

- Attacker can access the seller dashboard and then can edit the items he is selling

Recommendations

- The developer should disable these confidential default pages

References

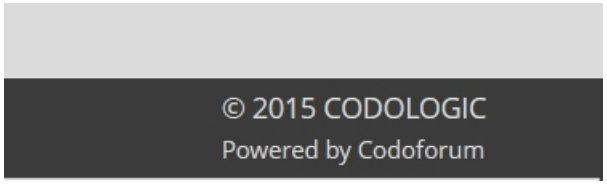
- https://www.indusface.com/blog/owasp-security-misconfiguration/
- https://hdivsecurity.com/owasp-security-misconfiguration

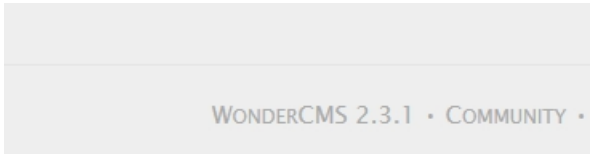
6.Components with known vulnerability

Components with known vulnerability	The urls given below are of the components with known vulnerability • Affected url http://13.235.132.114/wondercms/ http://13.235.132.114/forum/ **And PHP**

Observations

- I checked the versions of these components they were out dated





Observations

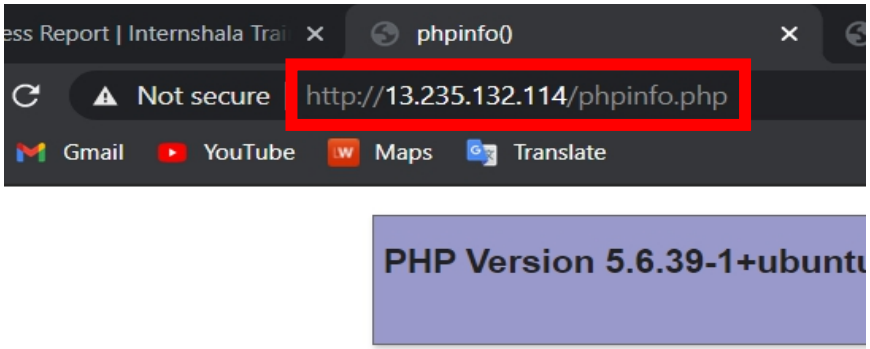
In 2015 version of codoforum was 3.0
now,

Key Facts

CMS name	**WonderCMS**
User rating	(77 votes, average: 4.18 out of 5) **WonderCMS** 4.18 5 77 Loading...
CMS Categories	CMS / Portals, Lite / Simple
Current **version** (stable)	2.5.1
Latest release date (stable)	05/03/2018

Observations

- The php version of this website is 5.6.39-1 which is out dated

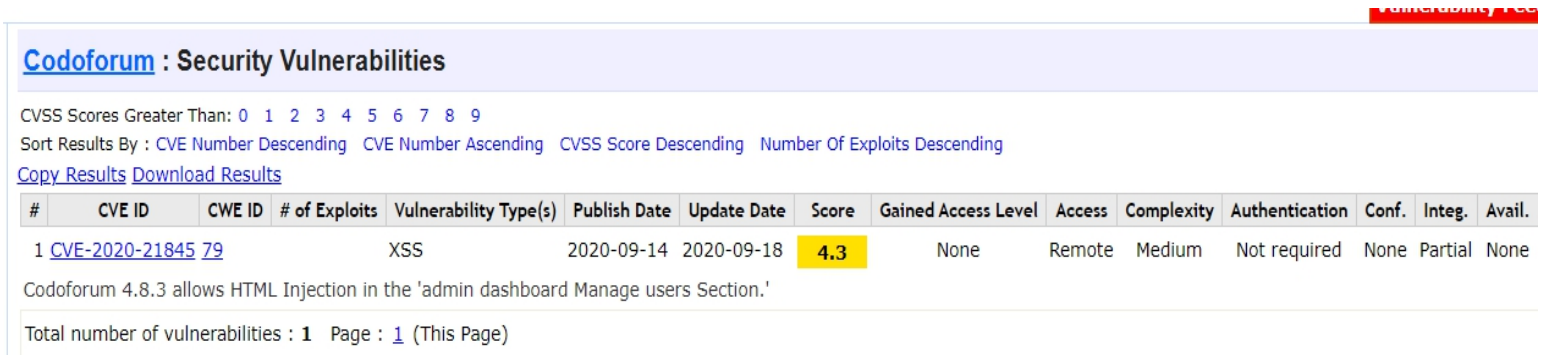


PHP.

Developer	The **PHP** Development Team, Zend Technologies
First appeared	1995
Stable release	8.0.6 / 6 May **2021**

POC

- Both the components have known public exploits



Codoforum : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2020-21845	79		XSS	2020-09-14	2020-09-18	4.3	None	Remote	Medium	Not required	None	Partial	None

Codoforum 4.8.3 allows HTML Injection in the 'admin dashboard Manage users Section.'

Total number of vulnerabilities : **1** Page : 1 (This Page)

POC

Wondercms » Wondercms : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2020-35314	78		Exec Code	2021-04-20	2021-04-23	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
A remote code execution vulnerability in the installUpdateThemePluginAction function in index.php in WonderCMS 3.1.3, allows remote attackers to upload a custom plugin which can contain arbitrary code and obtain a webshell via the theme/plugin installer.														
2	CVE-2020-35313	918		Exec Code	2021-04-20	2021-04-23	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
A server-side request forgery (SSRF) vulnerability in the addCustomThemePluginRepository function in index.php in WonderCMS 3.1.3 allows remote attackers to execute arbitrary code via a crafted URL to the theme/plugin installer.														
3	CVE-2020-29469	79		XSS	2020-12-30	2021-01-04	3.5	None	Remote	Medium	???	None	Partial	None
WonderCMS 3.1.3 is affected by cross-site scripting (XSS) in the Menu component. This vulnerability can allow an attacker to inject the XSS payload in the Setting - Menu and each time any user will visits the website directory, the XSS triggers and attacker can steal the cookie according to the crafted payload.														
4	CVE-2020-29247	79		XSS	2020-12-24	2021-04-22	3.5	None	Remote	Medium	???	None	Partial	None
WonderCMS 3.1.3 is affected by cross-site scripting (XSS) in the Admin Panel. An attacker can inject the XSS payload in Page keywords and each time any user will visit the website, the XSS triggers, and the attacker can able to steal the cookie according to the crafted payload.														
5	CVE-2020-29233	79		XSS	2020-12-30	2021-01-04	3.5	None	Remote	Medium	???	None	Partial	None
WonderCMS 3.1.3 is affected by cross-site scripting (XSS) in the Page description component. This vulnerability can allow an attacker to inject the XSS payload in the Page description and each time any user will visits the website, the XSS triggers and attacker can steal the cookie according to the crafted payload.														
6	CVE-2019-5956	22		Dir. Trav.	2019-09-12	2019-09-13	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Directory traversal vulnerability in WonderCMS 2.6.0 and earlier allows remote attackers to delete arbitrary files via unspecified vectors.														
7	CVE-2018-1000062	79		XSS	2018-02-09	2018-03-05	3.5	None	Remote	Medium	???	None	Partial	None
WonderCMS version 2.4.0 contains a Stored Cross-Site Scripting on File Upload through SVG vulnerability in uploadFileAction(), 'svg' => 'image/svg+xml' that can result in An attacker can execute arbitrary script on an unsuspecting user's browser. This attack appear to be exploitable via Crafted SVG File.														
8	CVE-2018-14387	384			2018-07-18	2018-09-19	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
An issue was discovered in WonderCMS before 2.5.2. An attacker can create a new session on a web application and record the associated session identifier. The attacker then causes the victim to authenticate against the server using the same session identifier. The attacker can access the user's account through the active session. The Session Fixation attack fixes a session on the victim's browser, so the attack starts before the user logs in.														
9	CVE-2018-7172	22		Dir. Trav.	2018-02-27	2018-03-23	5.5	None	Remote	Low	???	None	Partial	Partial
In index.php in WonderCMS before 2.4.1, remote attackers can delete arbitrary files via directory traversal.														
10	CVE-2017-14523	74			2018-01-26	2019-04-30	5.0	None	Remote	Low	Not required	None	Partial	None
** DISPUTED ** WonderCMS 2.3.1 is vulnerable to an HTTP Host header injection attack. It uses user-entered values to redirect pages. NOTE: the vendor reports that exploitation is unlikely because the attack can only come from a local machine or from the administrator as a self attack.														

PCO

11	CVE-2017-14522	79	XSS	2018-01-26	2019-03-13	4.3	None	Remote	Medium	Not required	None	Partial	None
** DISPUTED ** In WonderCMS 2.3.1, the application's input fields accept arbitrary user input resulting in execution of malicious JavaScript. NOTE: the vendor disputes this issue stating that this is a feature that enables only a logged in administrator to write execute JavaScript anywhere on their website.													
12	CVE-2017-14521	434		2018-01-26	2019-04-26	6.5	None	Remote	Low	???	Partial	Partial	Partial
In WonderCMS 2.3.1, the upload functionality accepts random application extensions and leads to malicious File Upload.													
13	CVE-2017-7951	352	CSRF	2017-04-21	2017-04-24	6.8	None	Remote	Medium	Not required	Partial	Partial	Partial
WonderCMS before 2.0.3 has CSRF because of lack of a token in an unspecified context.													
14	CVE-2014-8705	20	Exec Code File Inclusion	2017-03-17	2017-03-20	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
PHP remote file inclusion vulnerability in editInplace.php in Wonder CMS 2014 allows remote attackers to execute arbitrary PHP code via a URL in the hook parameter.													
15	CVE-2014-8704	22	Dir. Trav.	2017-03-17	2017-03-20	7.5	None	Remote	Low	Not required	Partial	Partial	Partial
Directory traversal vulnerability in index.php in Wonder CMS 2014 allows remote attackers to include and execute arbitrary local files via a crafted theme.													
16	CVE-2014-8703	79	XSS	2017-03-17	2017-03-20	4.3	None	Remote	Medium	Not required	None	Partial	None
Cross-site scripting (XSS) vulnerability in Wonder CMS 2014 allows remote attackers to inject arbitrary web script or HTML.													
17	CVE-2014-8702	200	+Info	2017-03-17	2017-03-31	5.0	None	Remote	Low	Not required	Partial	None	None
Wonder CMS 2014 allows remote attackers to obtain sensitive information by logging into the application with an array for the password, which reveals the installation path in an error message.													
18	CVE-2014-8701	200	+Info	2017-03-17	2017-03-20	5.0	None	Remote	Low	Not required	Partial	None	None
Wonder CMS 2014 allows remote attackers to obtain sensitive information by viewing /files/password, which reveals the unsalted MD5 hashed password.													
19	CVE-2011-5317	79	XSS	2015-01-01	2015-01-02	4.3	None	Remote	Medium	Not required	None	Partial	None
Cross-site scripting (XSS) vulnerability in editText.php in WonderCMS before 0.4 allows remote attackers to inject arbitrary web script or HTML via the content parameter.													

POC

- The running php version has multiple vulnerabilities

According to its banner, the version of PHP running on the remote web server is 5.6.x prior to 5.6.39. It is, therefore, affected by multiple vulnerabilities:

- An arbitrary command injection vulnerability exists in the imap_open function due to improper filters for mailbox names prior to passing them to rsh or ssh commands. An authenticated, remote attacker can exploit this by sending a specially crafted IMAP server name to cause the execution of arbitrary commands on the target system. (CVE-2018-19518)
- A denial of service (DoS) vulnerability exists in ext/imap/php_imap.c. An unauthenticated, remote attacker can exploit this issue, via an empty string in the message argument to the imap_mail function, to cause the application to stop responding. (CVE-2018-19935)
- A heap buffer over-read exists in the phar_parse_pharfile function. An unauthenticated, remote attacker can exploit this to read

Business impact- Extremely high

- Anyone can perform any attacks (available) as all the exploits are available publicly .
- It can cause severe damage to the website
- He may be able to upload backdoor shells
- He will easily deface your website

Recommendations

- Update all the components and the php version which is running on it
- Hide the current versions info from there pages

References

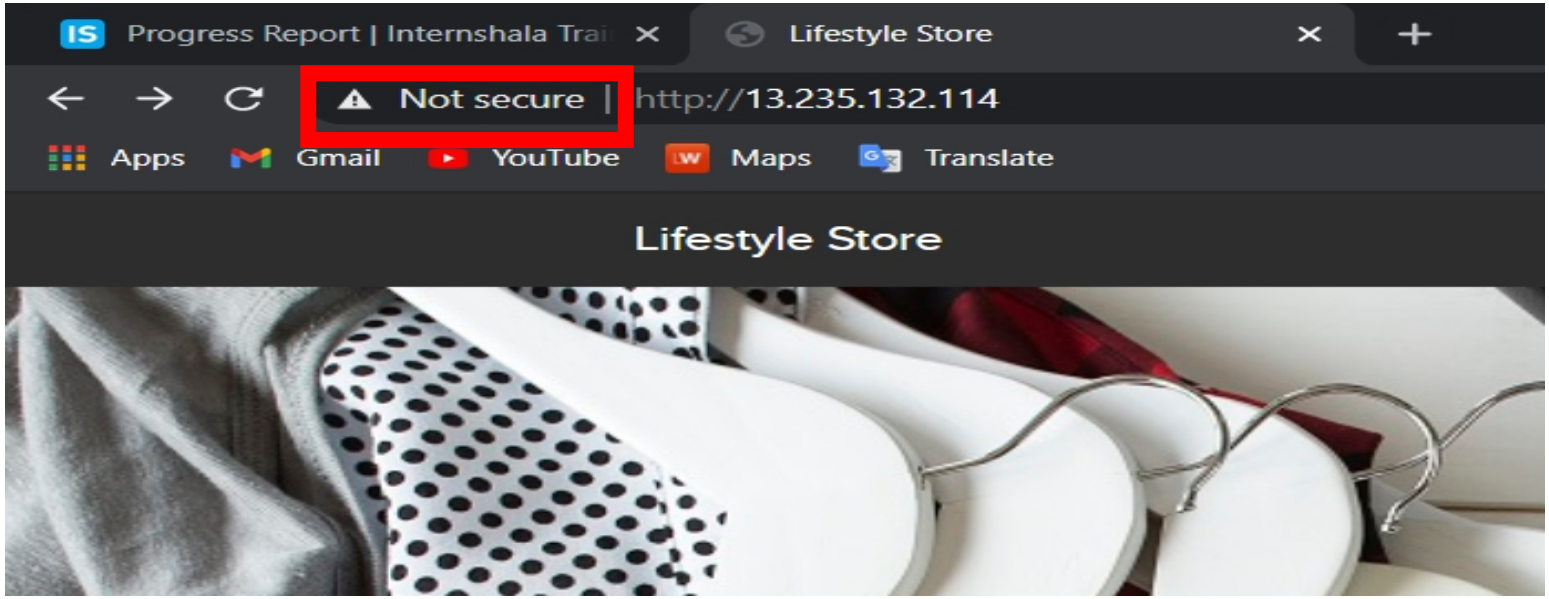
- https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A9-Using_Components_with_Known_Vulnerabilities
- https://vulners.com/nessus/PHP_5_6_39.NASL
- https://www.cvedetails.com/index.php

7.Crypto Configuration Flaws

Crypto Configuration Flaw	Crypto Configuration Flaws are found in the modules below. • Affected URL : http://13.235.132.114/(All the webpages ,blogs ,forum)

Observation

- Clearly ,all the webpages use 'http' and not 'https' which is far less secure and not encrypted.



Business Impact - High

- Security is almost halved in http providing easy man-in-the-middle attack and others which makes it easy for attacker to go through the data transmitted over the internet.

Recommendation

- Use https instead of http as the protocol

References

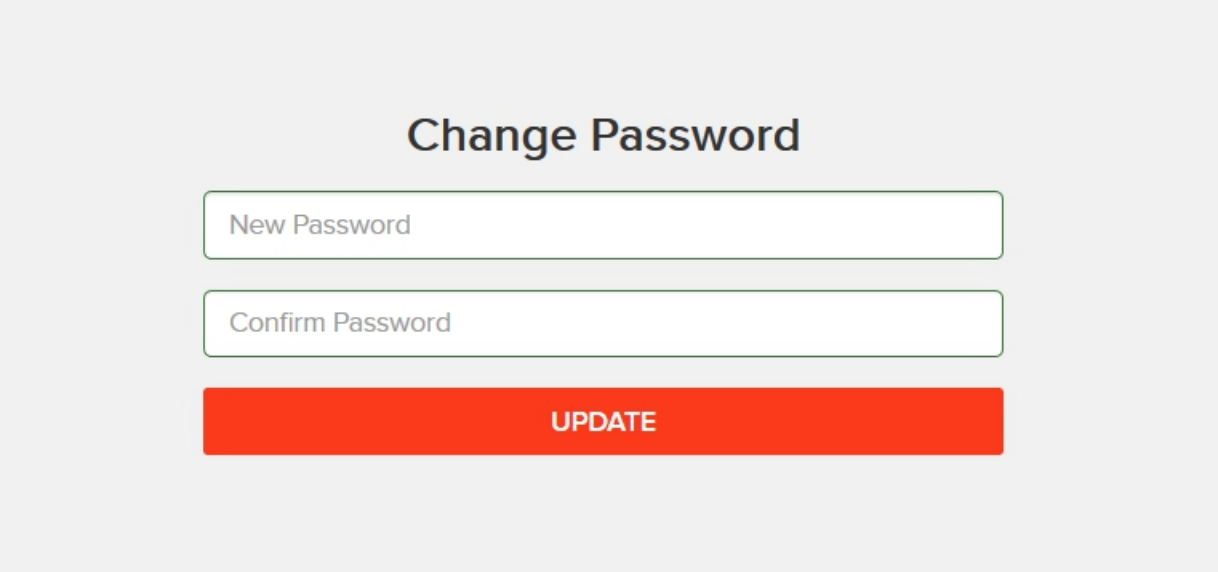
- https://www.owasp.org/index.php/Category:Cryptographic_Vulnerability
- https://www.w3.org/Protocols/rfc2616/rfc2616-sec15.html

8.C.S.R.F.

CSRF	The url given below is vulnerable to CSRF • Affected url http://13.235.132.114/profile/change_password.php http://13.235.132.114/cart/cart.php

observations

- There is a change password option in profile page



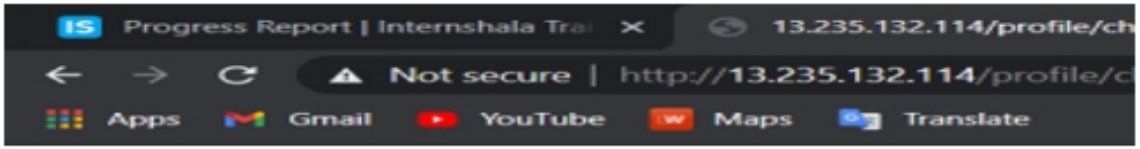
POC

- Make a html page to change username and password

```
<html>
<head>
<title> CSRF POC </title>
</head>
<body>
    <form name='change-password' id='change-password' method='POST' action='http://52.66.65.223/profile/change_password_submit.php'>
    <input type='password' placeholder="New Password" name="password" id="password" value="1234">
    <input type='password' placeholder="Confirm Password" name="password_confirm" id="password_confirm" value="1234">
    <button type='submit' class="btn btn-primary">Update</button>
</body>
</html>
```

POC

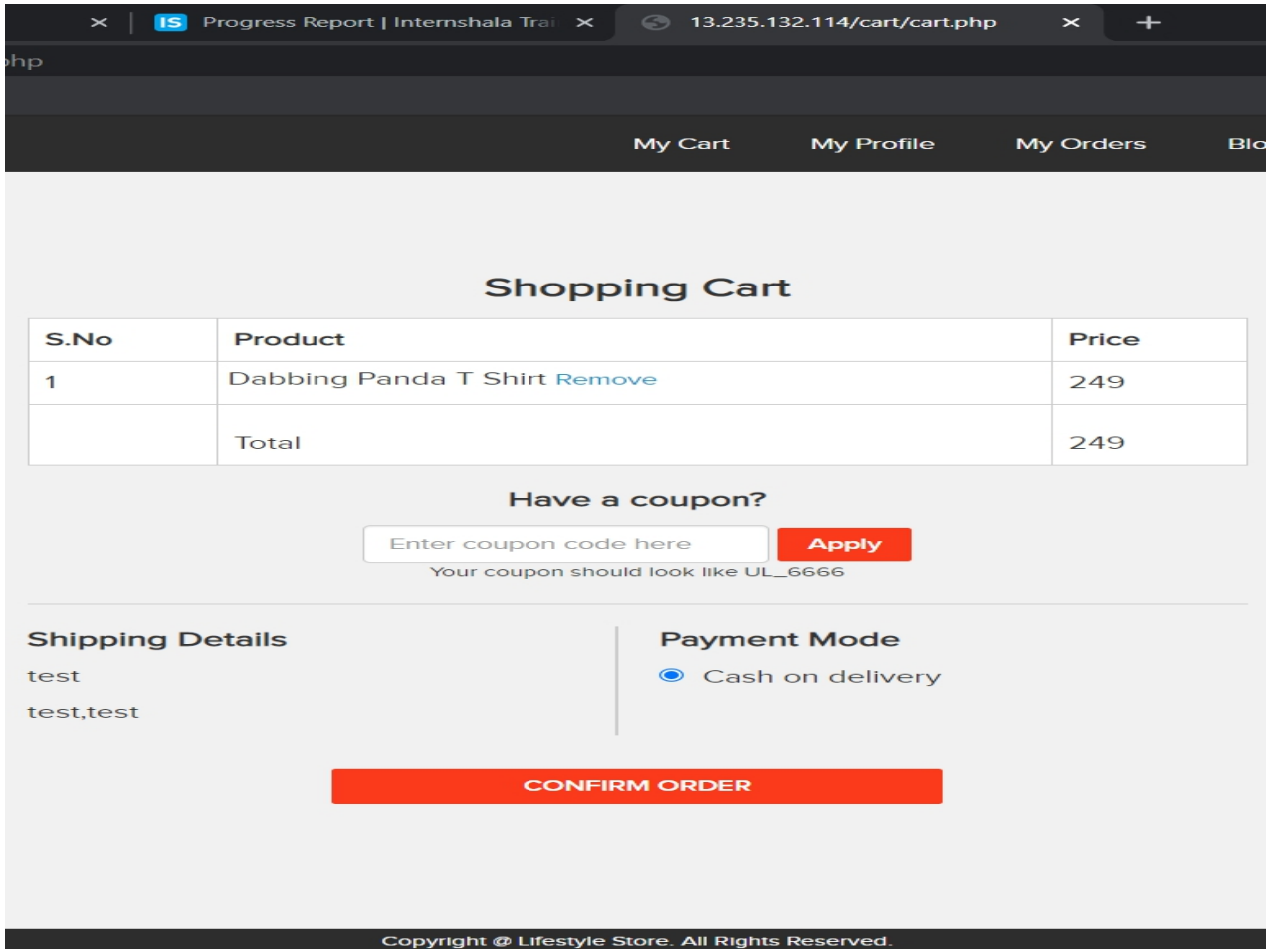
- On clicking the update button we get success



{"success":true,"successMessage":"Password updated succesfully."}

Observations

- There is a confirm button in my orders



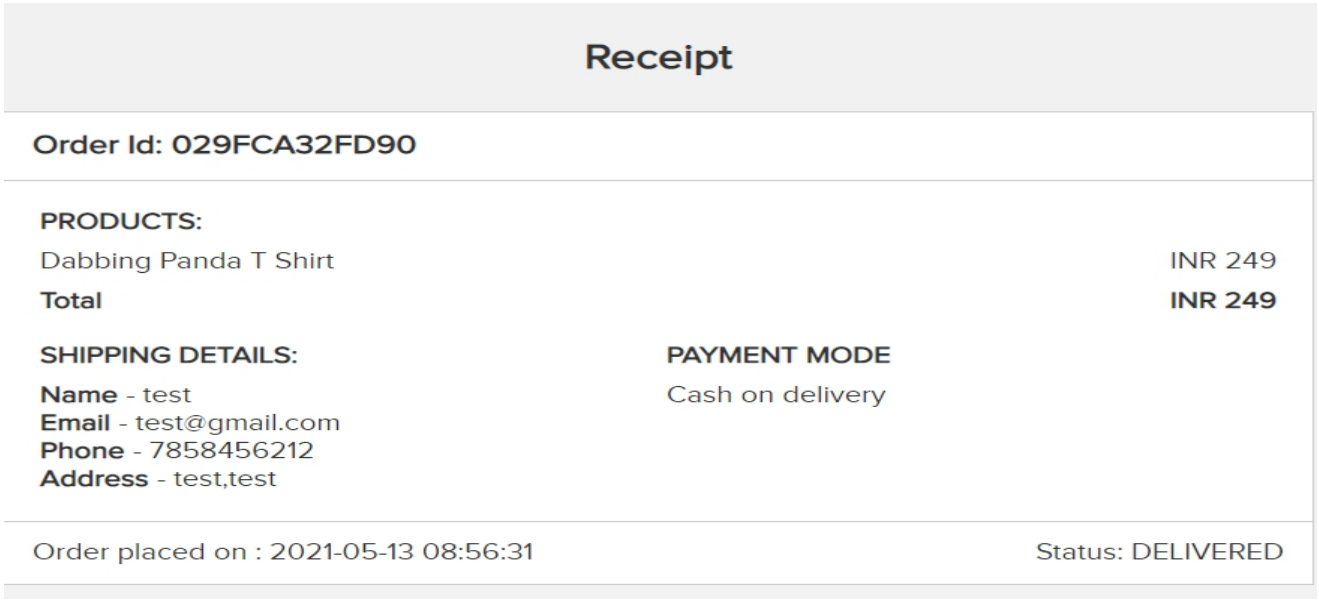
POC

- Make a html page to confirm order

```
<head>
<title> CSRF POC </title>
</head>
<body>
    <form action="http://52.66.65.223/orders/confirm.php" method='POST'>
    <input type='Submit' value="Submit Request"></input>
</body>
</html>
```

POC

- On executing the page order is confirmed



Receipt

Order Id: 029FCA32FD90

PRODUCTS:

Dabbing Panda T Shirt — INR 249

Total — **INR 249**

SHIPPING DETAILS:

Name - test
Email - test@gmail.com
Phone - 7858456212
Address - test,test

PAYMENT MODE

Cash on delivery

Order placed on : 2021-05-13 08:56:31

Status: DELIVERED

Business impact- severe

- Attacker can change the password by uploading phishing pages
- Attacker can confirm the order without consent of user

Recommendations

- Use of tokens and session cookies
- Referrer header should be checked at server side

References

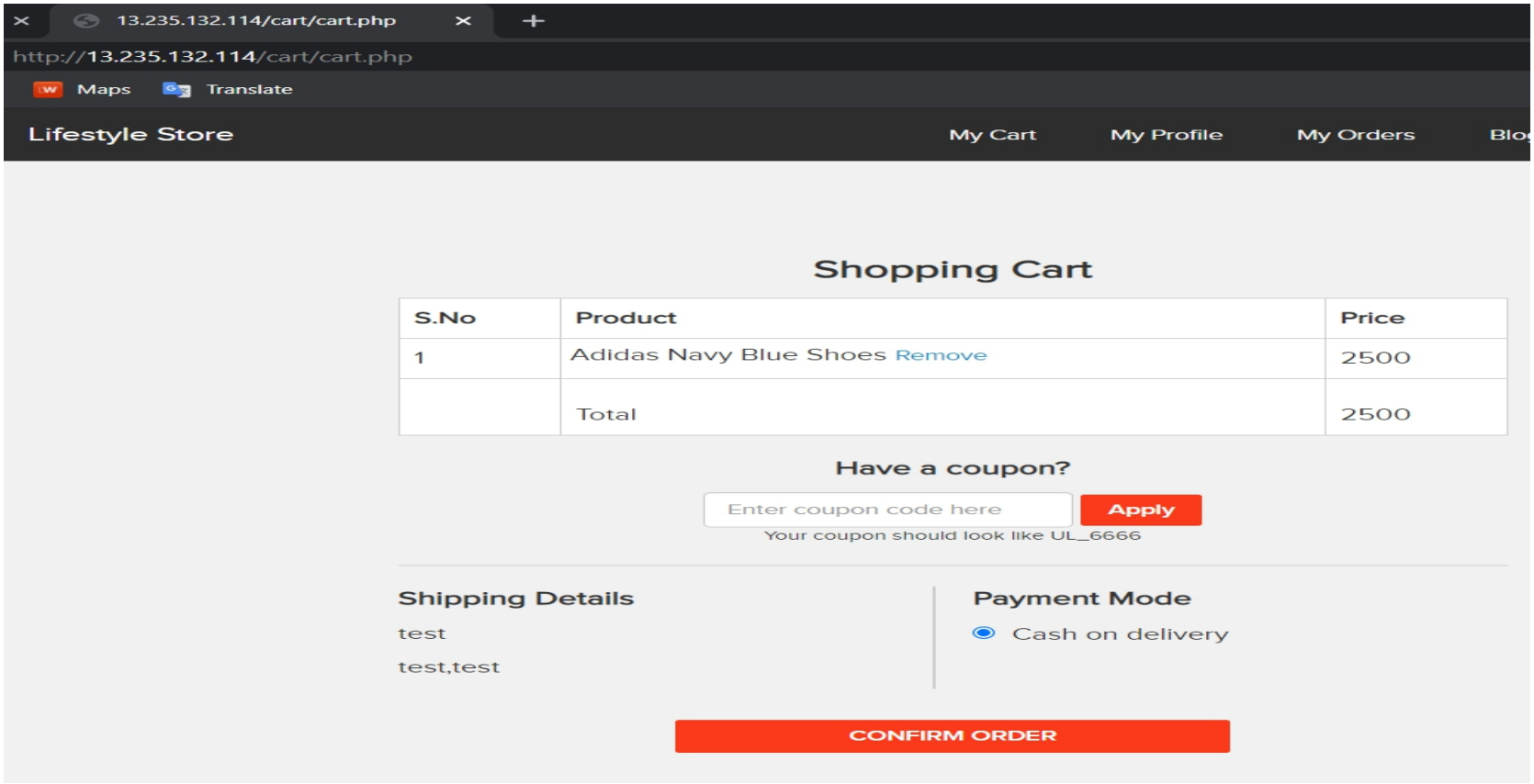
- https://owasp.org/www-community/attacks/csrf
- https://www.netsparker.com/blog/web-security/csrf-cross-site-request-forgery/

9.Coupon code brute forcing

Coupon code brute forcing	In the below url brute forcing can be performed for discounts • Affected url http://13.235.132.114/cart/apply_coupon.php

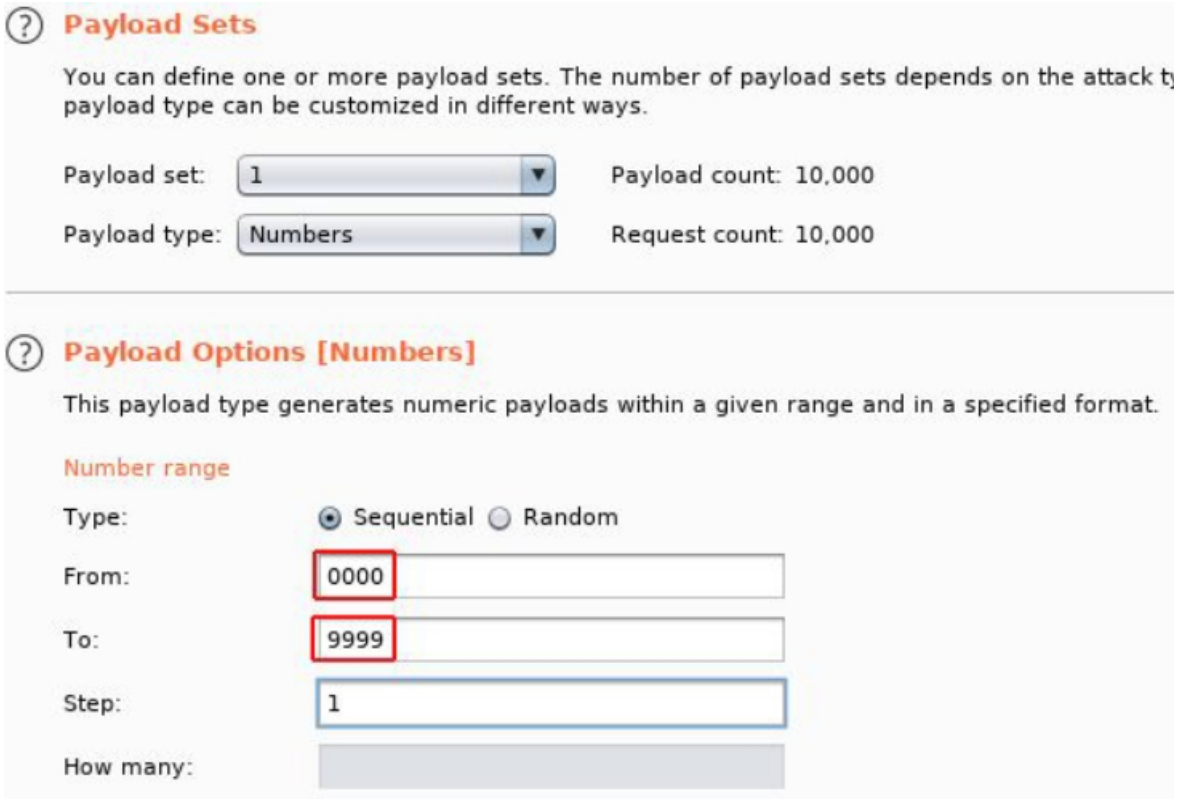
Observations

- When we go to the cart we see the apply coupon and coupon example



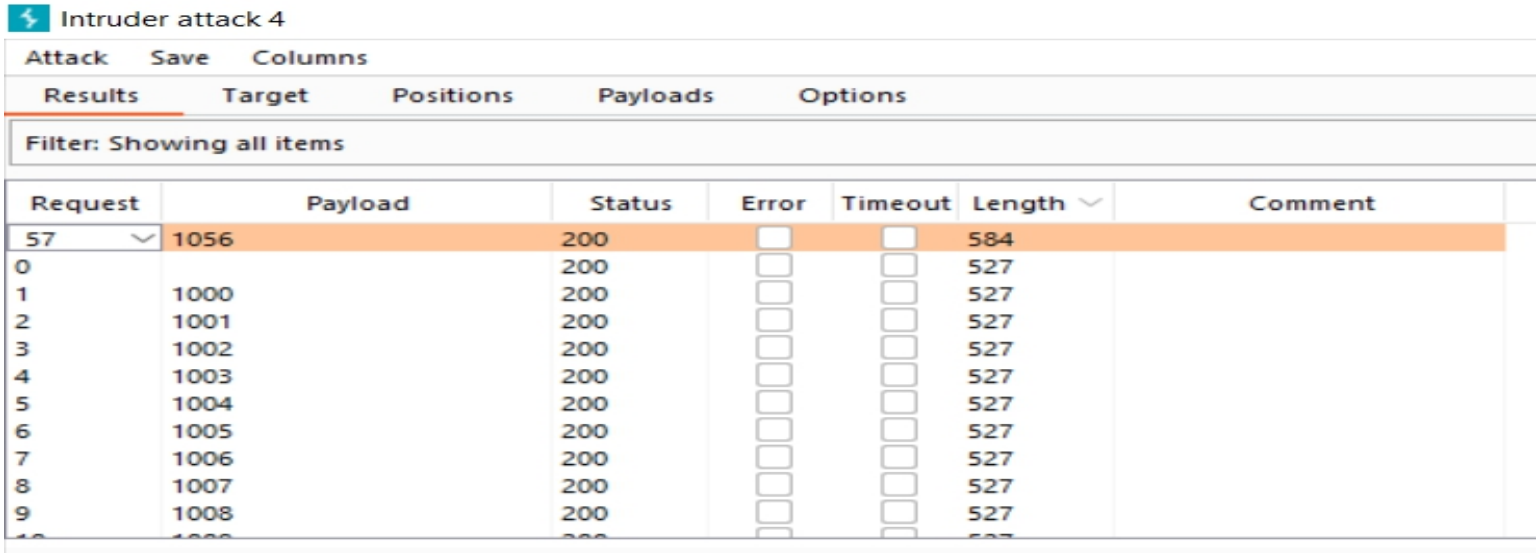
Observations

Brute forcing the coupon code

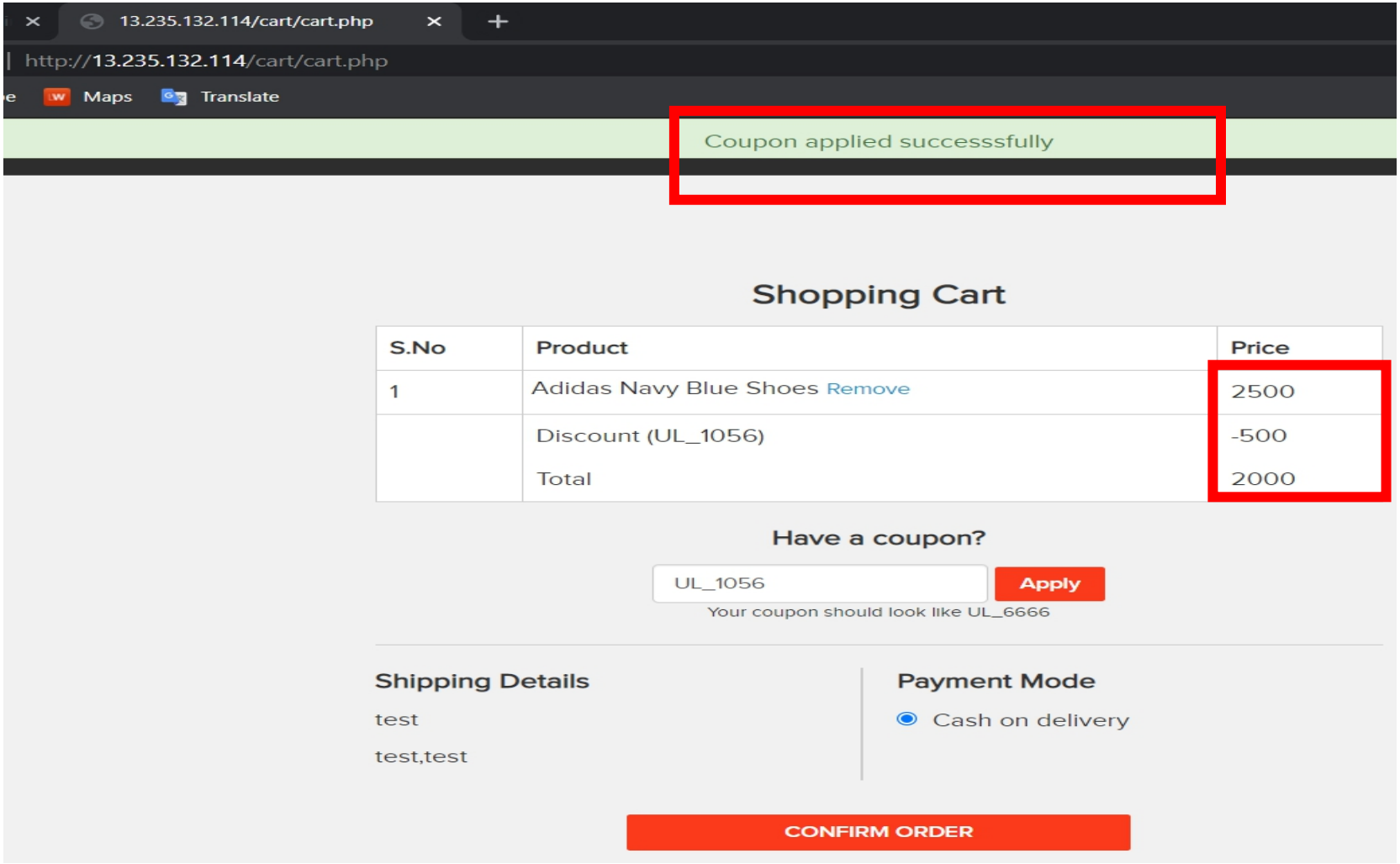


POC

- We were sucessful



POC


13.235.132.114/cart/cart.php
http://13.235.132.114/cart/cart.php
Maps
Translate
Coupon applied successsfully
Shopping Cart
S.No
Product
Price
1
Adidas Navy Blue Shoes Remove
2500
Discount (UL_1056)
-500
Total
2000
Have a coupon?
UL_1056
Apply
Your coupon should look like UL_6666
Shipping Details
test
test,test
Payment Mode
Cash on delivery
CONFIRM ORDER

Business impact - severe

- Attacker can easily order the items on extreme discounts which will be harmful for the company

Recommendation

- Coupon codes should have limited no of use and regenerated after sometime
- Coupon code should be random alpha-numeric characters

References

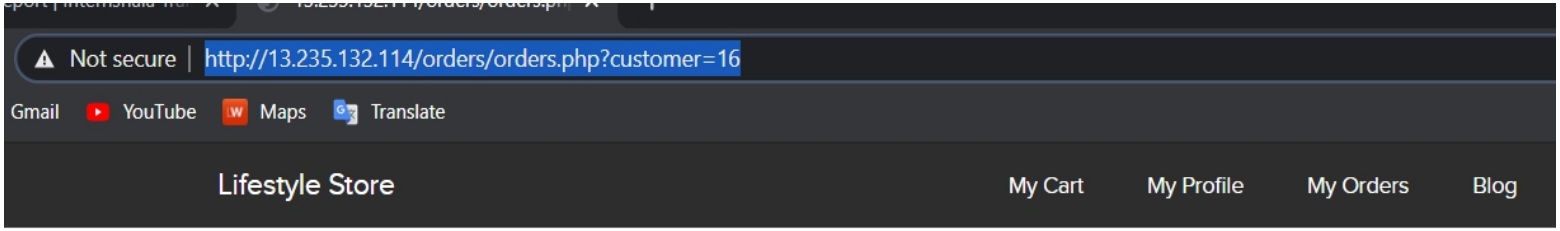
- https://www.digitalcommerce360.com/2017/03/17/prevent-fraud-brute-force-online-coupon-gift-card-attacks/
- https://www.couponxoo.com/brute-force-attack-coupon-code

10.Insecure direct object references

Insecure direct object references	The bellow mentioned url is vulnerable to IDOR • Affected url http://13.235.132.114/orders/orders.php?customer=(here) http://13.235.132.114/profile/(here)/edit/ http://13.235.132.114/orders/generate_receipt/ordered/13

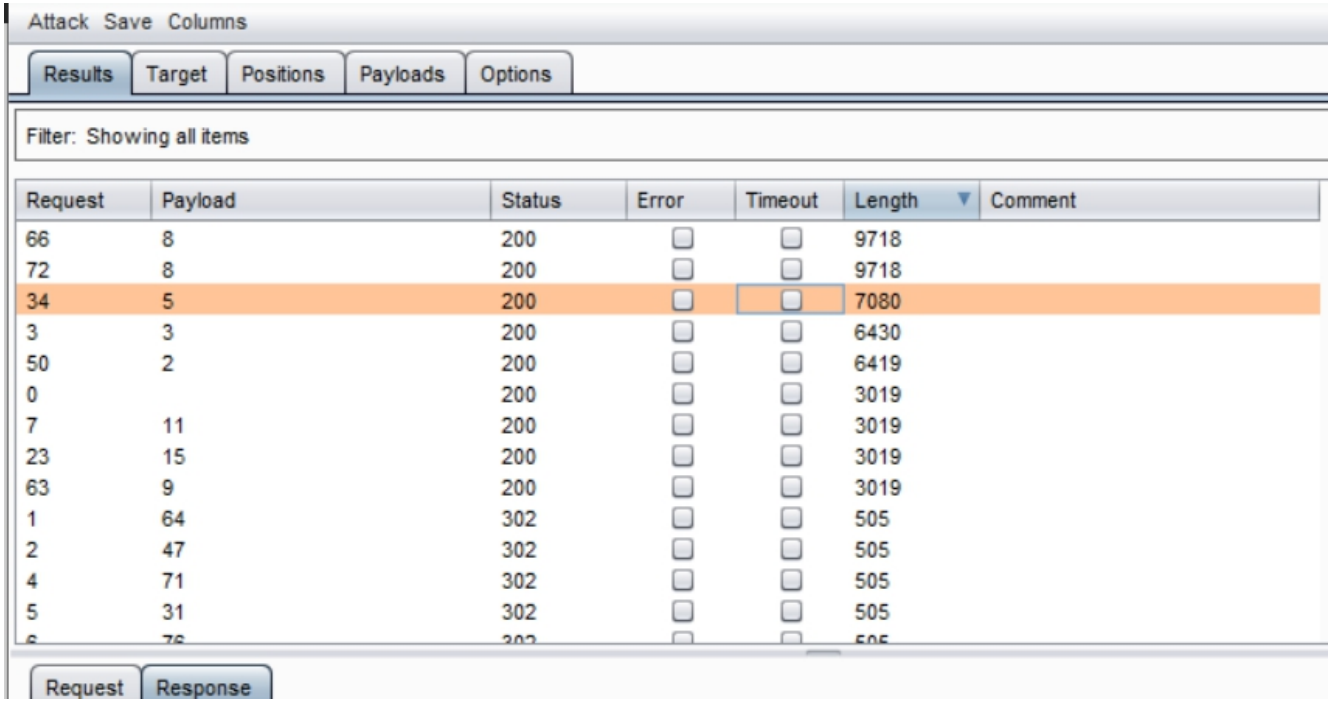
Observations

- In the my orders page I saw customer no in url

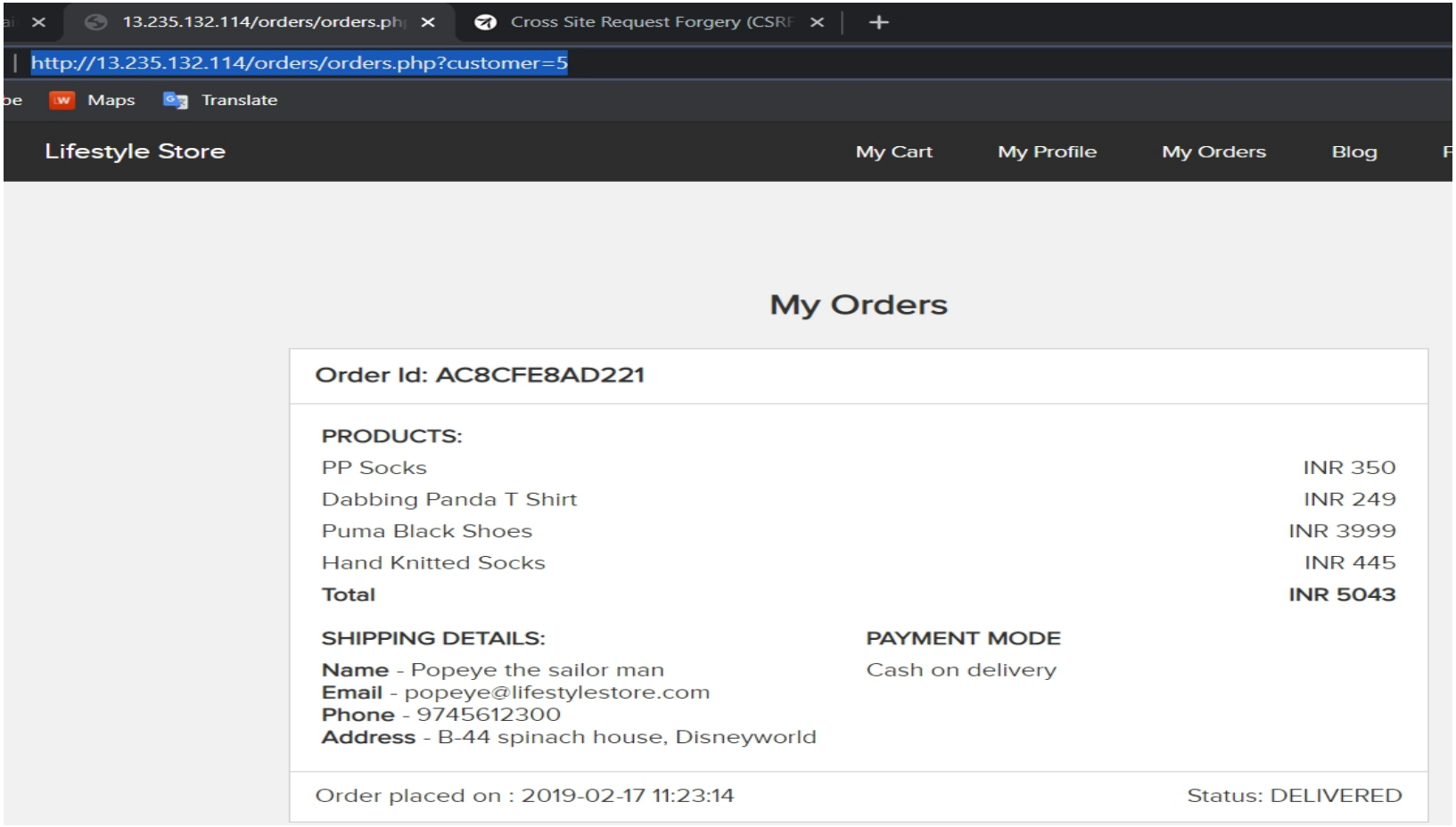


Observations

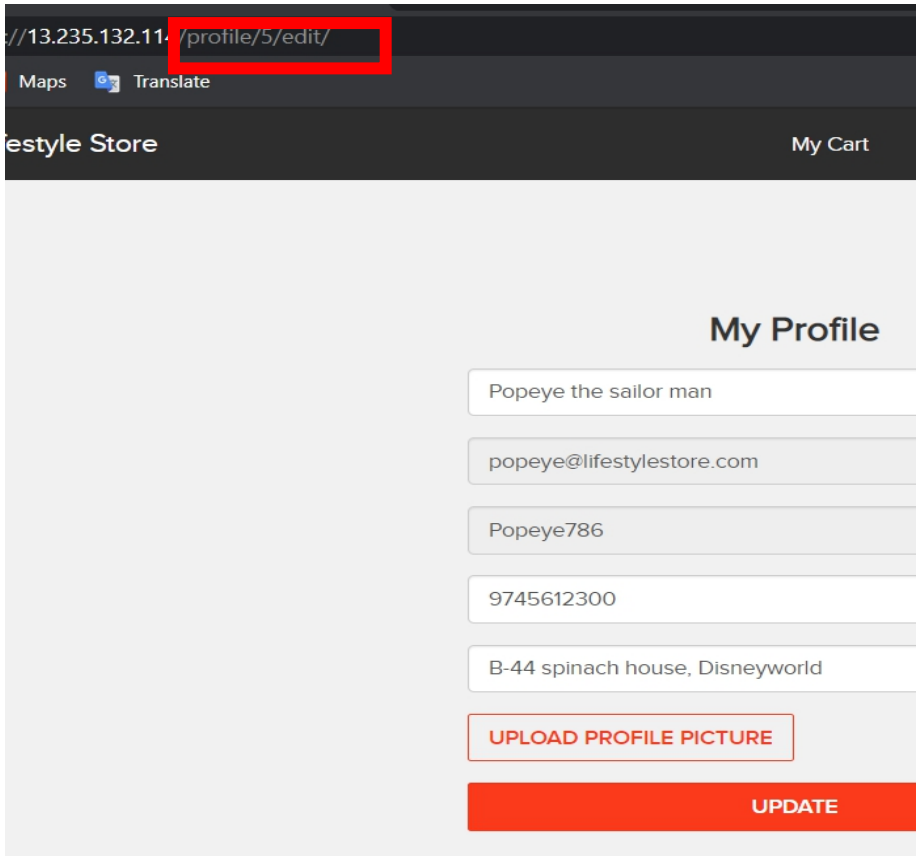
- I brute forced it

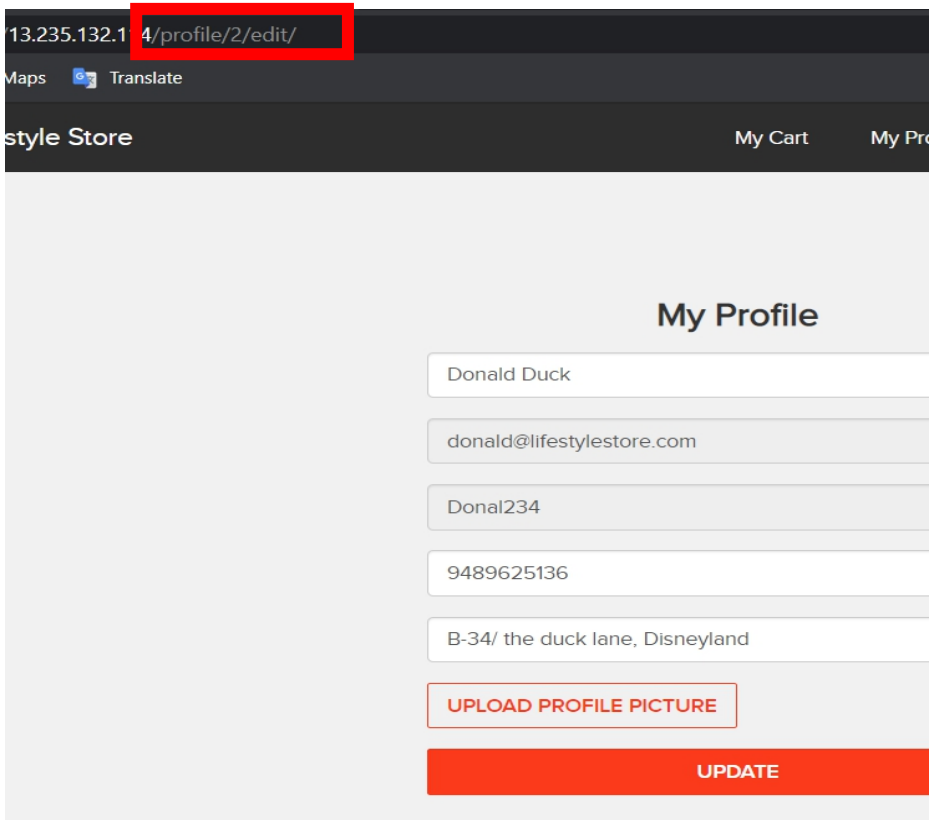


POC

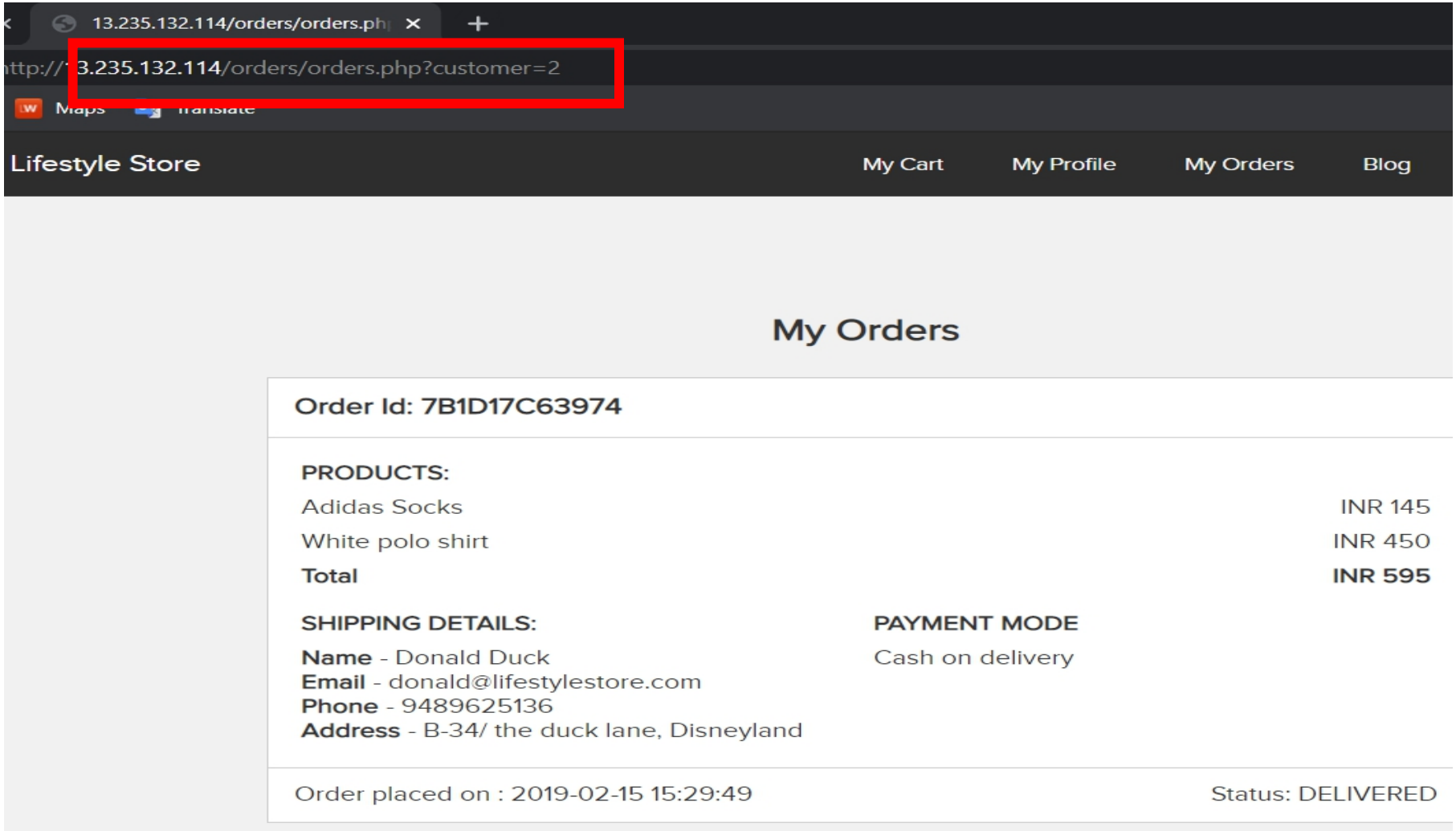

13.235.132.114/orders/orders.ph
Cross Site Request Forgery (CSRF
http://13.235.132.114/orders/orders.php?customer=5
Maps
Translate
Lifestyle Store
My Cart
My Profile
My Orders
Blog
My Orders
Order Id: AC8CFE8AD221
PRODUCTS:
PP Socks
INR 350
Dabbing Panda T Shirt
INR 249
Puma Black Shoes
INR 3999
Hand Knitted Socks
INR 445
Total
INR 5043
SHIPPING DETAILS:
Name - Popeye the sailor man
Email - popeye@lifestylestore.com
Phone - 9745612300
Address - B-44 spinach house, Disneyworld
PAYMENT MODE
Cash on delivery
Order placed on : 2019-02-17 11:23:14
Status: DELIVERED

POC


/profile/5/edit/
Maps
Translate
My Cart
My Profile
Popeye the sailor man
popeye@lifestylestore.com
Popeye786
9745612300
B-44 spinach house, Disneyworld
UPLOAD PROFILE PICTURE
UPDATE


/profile/2/edit/
Maps
Translate
My Cart
My Profile
Donald Duck
donald@lifestylestore.com
Donal234
9489625136
B-34/ the duck lane, Disneyland
UPLOAD PROFILE PICTURE
UPDATE

POC


13.235.132.114/orders/orders.ph
http://13.235.132.114/orders/orders.php?customer=2
Maps
Translate
Lifestyle Store
My Cart
My Profile
My Orders
Blog
My Orders
Order Id: 7B1D17C63974
PRODUCTS:
Adidas Socks
INR 145
White polo shirt
INR 450
Total
INR 595
SHIPPING DETAILS:
Name - Donald Duck
Email - donald@lifestylestore.com
Phone - 9489625136
Address - B-34/ the duck lane, Disneyland
PAYMENT MODE
Cash on delivery
Order placed on : 2019-02-15 15:29:49
Status: DELIVERED

Recomendations

- Instead of requiring the references in the URL, use the information already present in the user's session on the server to locate the resources to serve.
- If it is not possible to avoid exposing the references to objects in the URL, as explained earlier, the *indirect reference map* technique is helpful. The idea behind it is to substitute the sensitive direct internal reference in URL parameters or form fields with a random value that is difficult to predict (such as a GUID) or specific only to the logged-in user

References

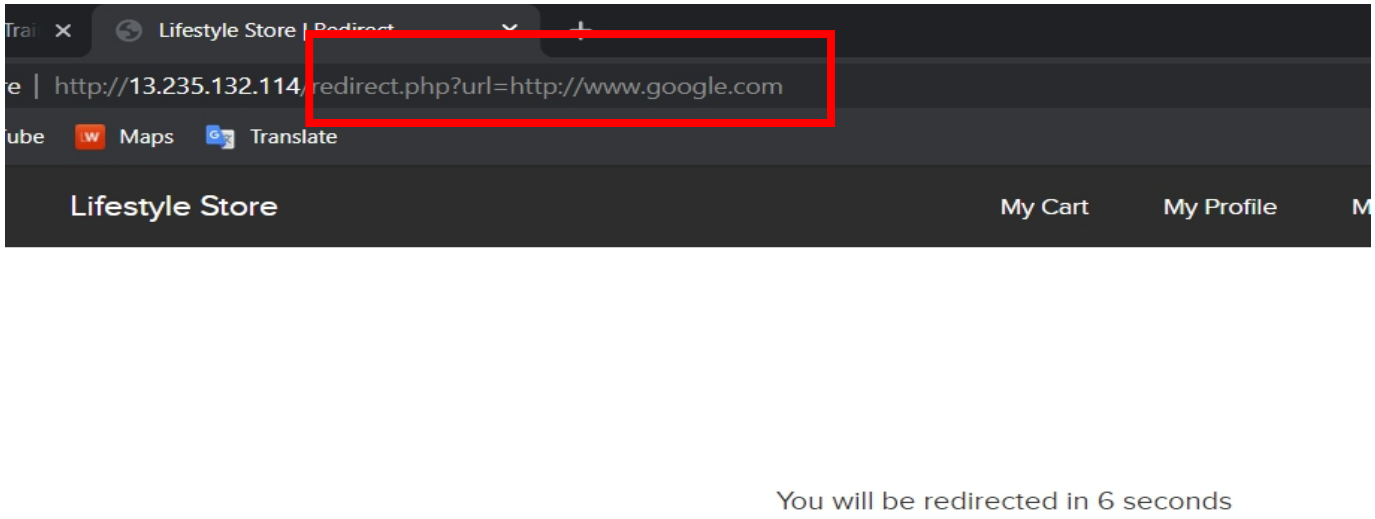
- https://www.oreilly.com/library/view/securing-node-applications/9781491982426/ch04.html
- https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object_Reference_Prevention_Cheat_Sheet.html

11.Open redirection

Open redirection	The url given below Is vulnerable to open redirection • Affected url http://13.235.132.114/redirect.php?url=(here) in the parentheses http://13.235.132.114/redirect.php?url=http://www.google.co.in

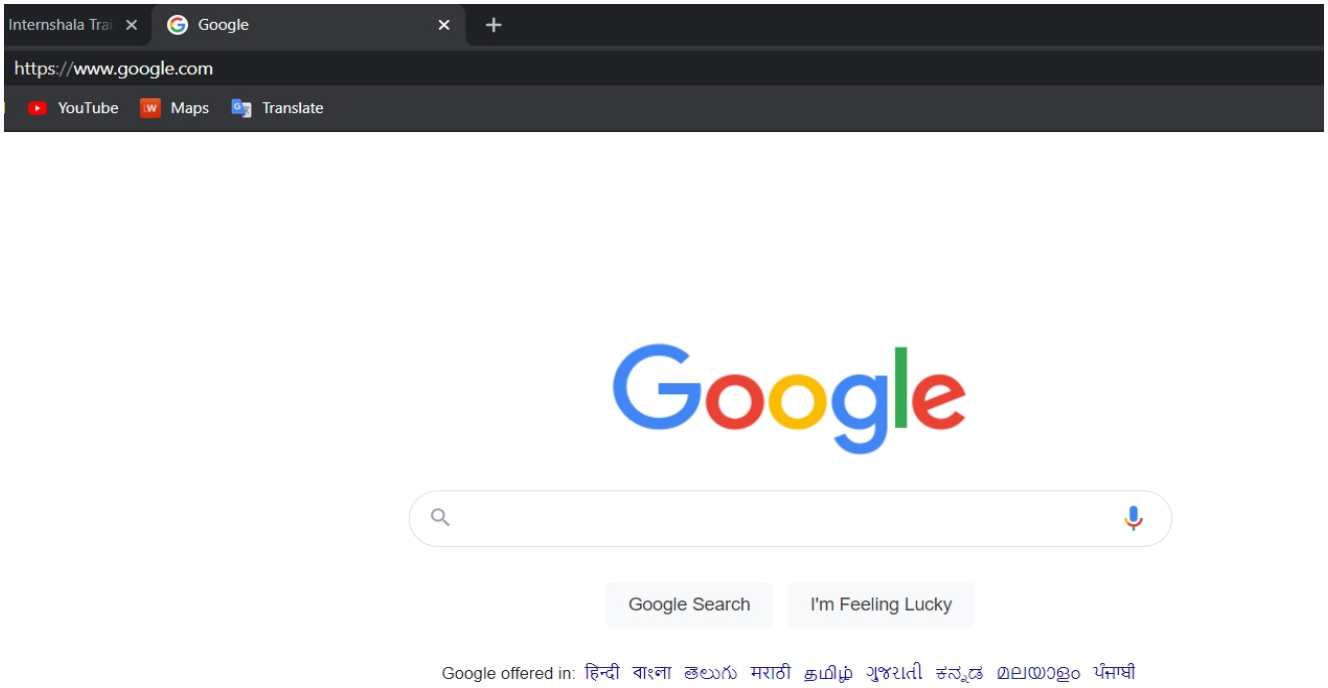
Observations

- On clicking the brand website redirection occurs



POC

- On changing the link to google.com we were redirected to it



Business impact- severe

- He can access the users personnel credentials which would be very harmful
- They can redirect your page to a malware site
- They can redirect you to phishing pages

Recommendations

- Design your app to avoid URL redirects or forwards as a best practice. If unavoidable, encrypt the target URL such that the URL:token mapping is validated on the server.
- Verify URL patterns using regular expressions to check if they belong to valid URLs. However, malicious URLs can pass that check.

References

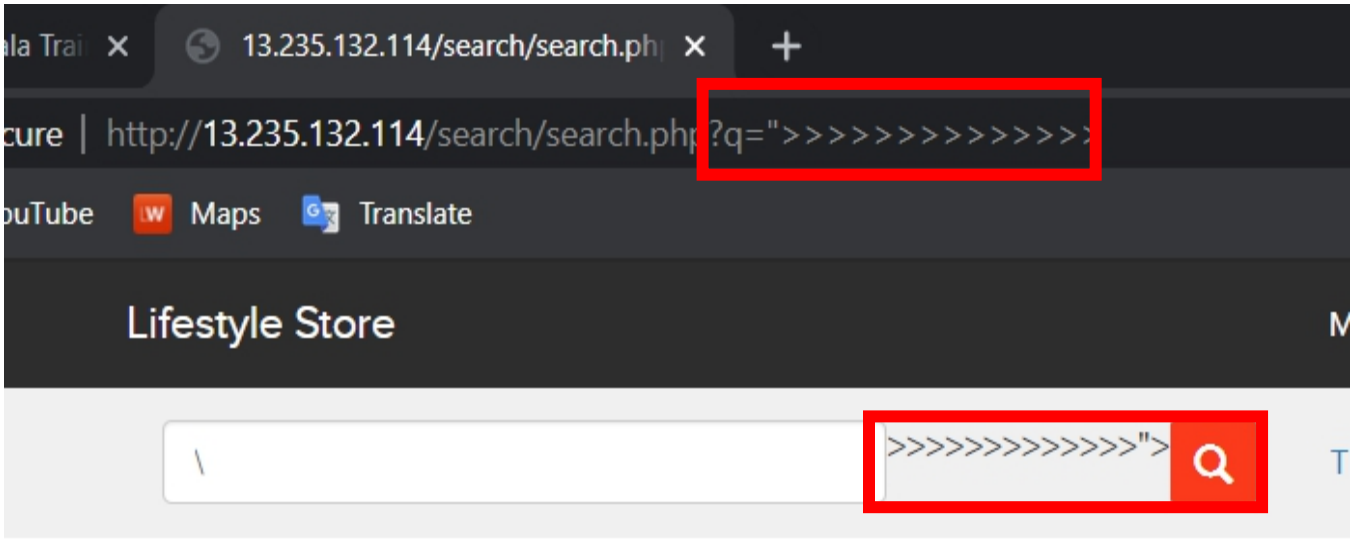
- https://spanning.com/blog/open-redirection-vulnerability-web-based-application-security-part-1/#:~:text=Understanding%20the%20Unvalidated%20Redirects%20Vulnerability&text=However%2C%20it%20can%20be%20misused,data%20and%20credibility%20into%20jeopardy.
- https://www.netsparker.com/blog/web-security/open-redirection-vulnerability-information-prevention/

12.Cross site scripting

Cross site scripting	The below mentioned urls are vulnerable to temporary and stored XSS • Affected urls Temporary -http://13.235.132.114/search/search.php?q=(here) Stored- http://13.235.132.114/products/details.php?p_id=(here)

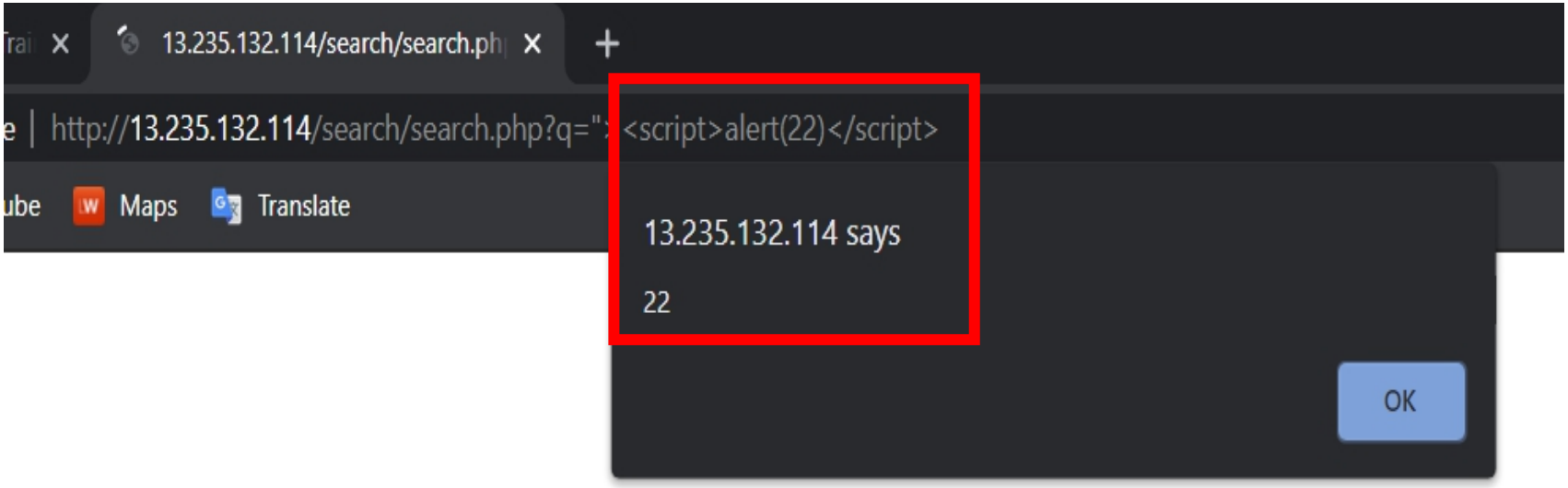
Observations

- In the search bar when I entered “> I found this



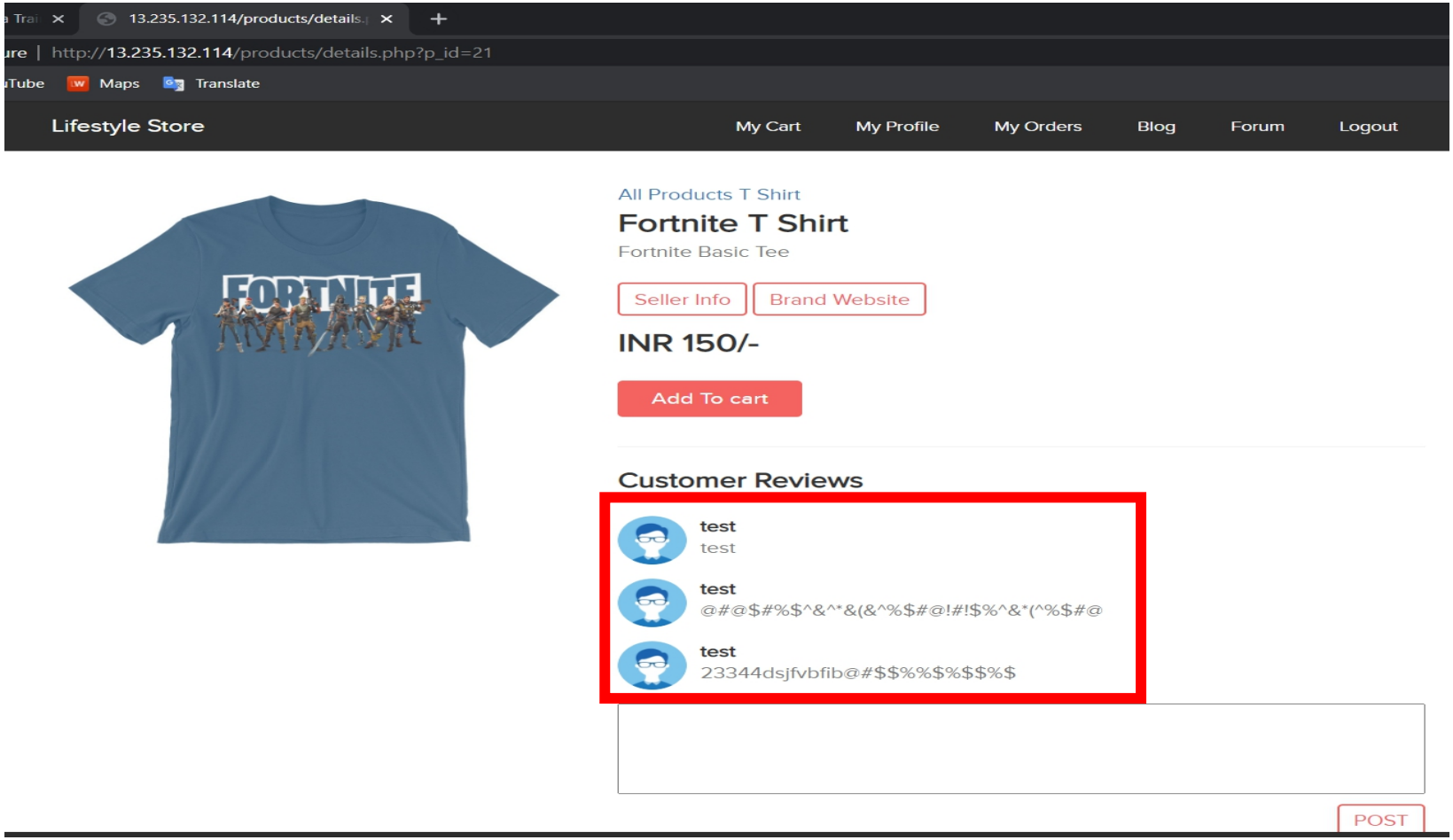
POC

- When I entered the script popup code it was executed



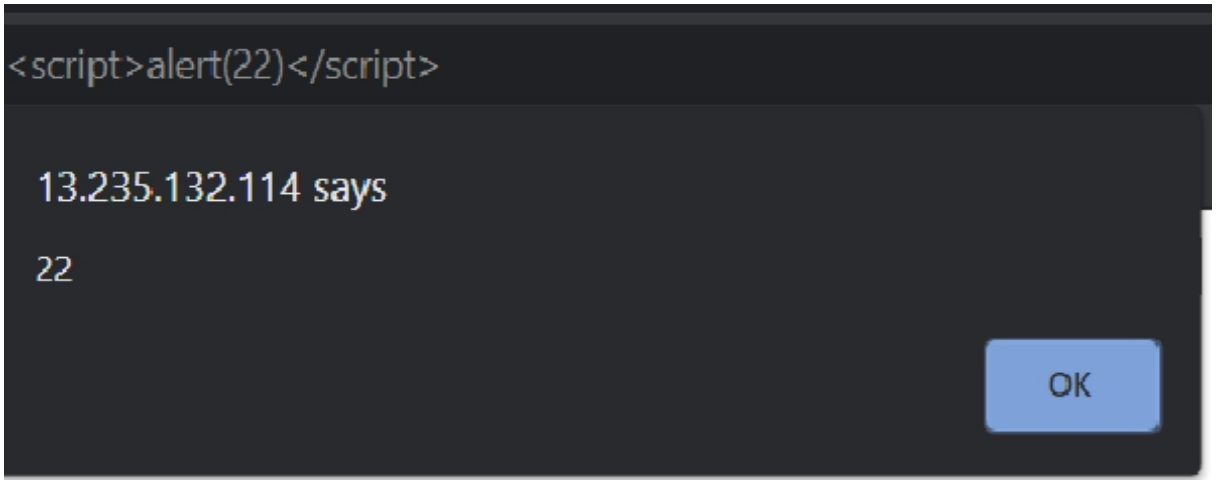
Observations

- In the comment section of every product items the comment was stored



POC

- When I entered the script pop up code it was executed and stored



Business impact- severe

- Hacker can access any user credentials by injecting ***malicious*** scripts
- He can even change the html format of website

Recommendations

- By escaping user input. Escaping data means taking the data an application has received and ensuring it’s secure before rendering it for the end user
- Validating input is the process of ensuring an application is rendering the correct data and preventing malicious data from doing harm to the site, database, and users
- A third way to prevent cross-site scripting attacks is to sanitize user input. Sanitizing data is a strong defense, but should not be used alone to battle XSS attacks

References

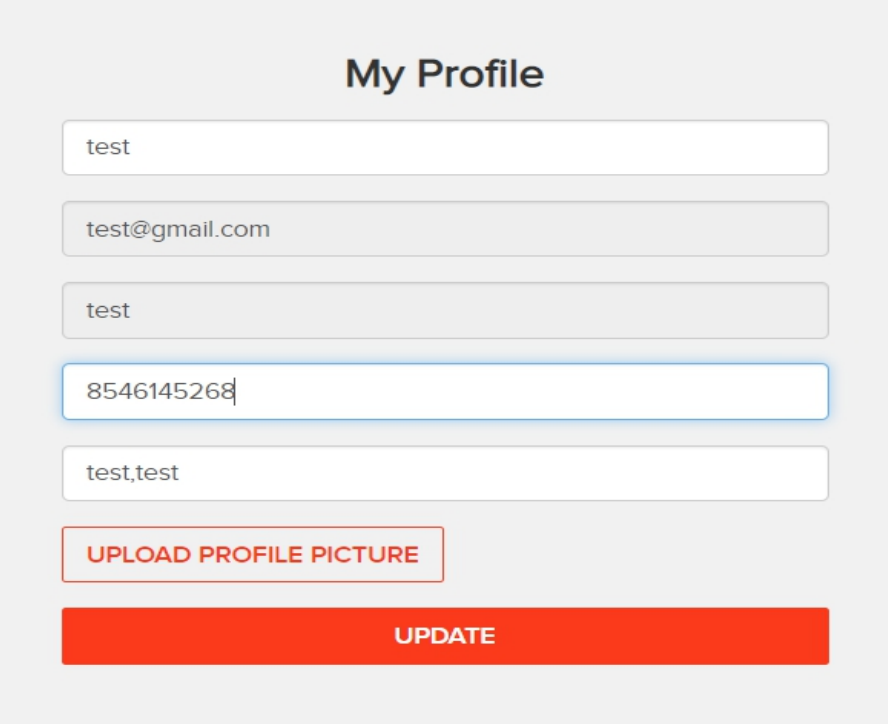
- https://www.checkmarx.com/2017/10/09/3-ways-prevent-xss/
- https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html
- https://owasp.org/www-community/attacks/xss/

13.Client side filter bypass

Client side filter bypass	The url given below is vulnerable to client side filter bypass • Afected url http://13.235.132.114/profile/16/edit/

Observations

- After changing the information I was able to change it again via client side filter bypass

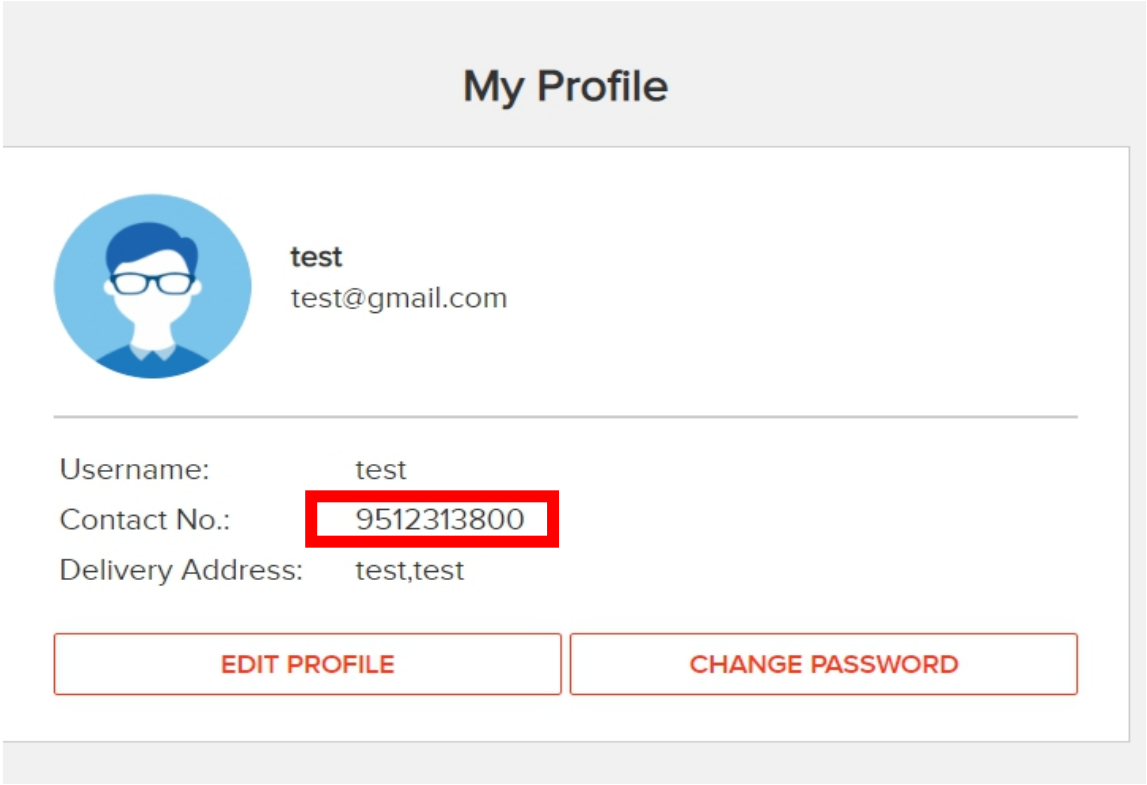


POC

```
POST /profile/submit.php HTTP/1.1
Host: 13.235.132.114
Content-Length: 605
Accept: text/plain, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.93 Safari/537.36
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryo3Px2ldXCuUVK516
Origin: http://13.235.132.114
Referer: http://13.235.132.114/profile/16/edit/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: key=C9F21D01-7EC2-B6E0-6528-3C2B097C41BB; PHPSESSID=78113osg2i7o8orukrg48p9805; X-XSRF-TOKEN=62c956821efc733718c99f78d8927b80a85cd50240e38200ede80d76f0e08342
Connection: close

------WebKitFormBoundaryo3Px2ldXCuUVK516
Content-Disposition: form-data; name="name"

test
------WebKitFormBoundaryo3Px2ldXCuUVK516
Content-Disposition: form-data; name="contact"

9512313800
------WebKitFormBoundaryo3Px2ldXCuUVK516
Content-Disposition: form-data; name="address"

test,test
------WebKitFormBoundaryo3Px2ldXCuUVK516
Content-Disposition: form-data; name="user_id"

16
------WebKitFormBoundaryo3Px2ldXCuUVK516
Content-Disposition: form-data; name="X-XSRF-TOKEN"

62c956821efc733718c99f78d8927b80a85cd50240e38200ede80d76f0e08342
------WebKitFormBoundaryo3Px2ldXCuUVK516--
```

POC

- Original info was changed



Business impact-high

- This would only trouble the users which will be giving bad feed back on you website

Recommendations

- Cookies should be used .
- Referrer headers should be used
- Proper security checks should be done

References

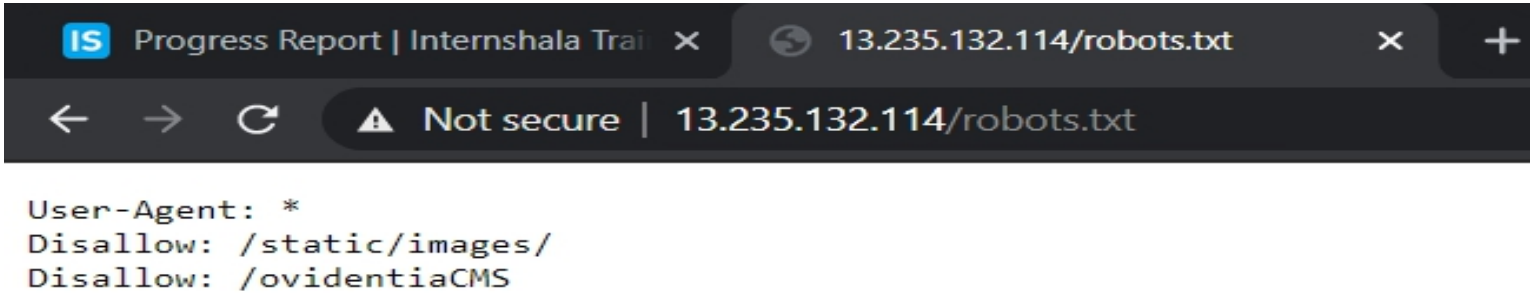
- https://portswigger.net/support/using-burp-to-bypass-client-side-javascript-validation
- https://www.slideshare.net/SamBowne/cnit-129s-ch-5-bypassing-clientside-controls

14.Directory listing

Directory listing	The url given below is listing the directories • Affected url http://13.235.132.114/static/images/

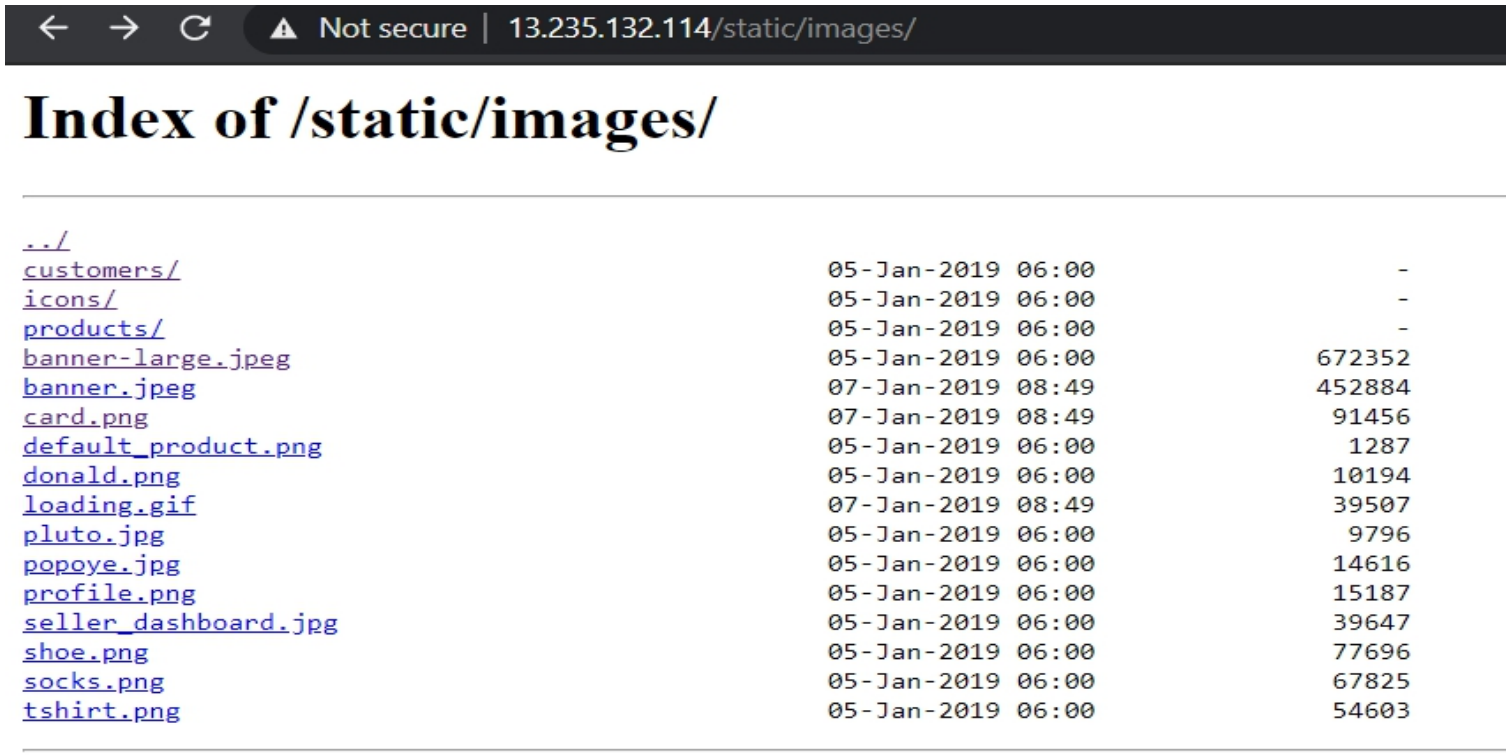
Observations

- In robots.txt file I found static/images/



POC

- Listed directories



Business impact-high

- These directories will be useful for the attacker to collect information about the website

To plan a attack

Recommendations

- Disable these listed directories

References

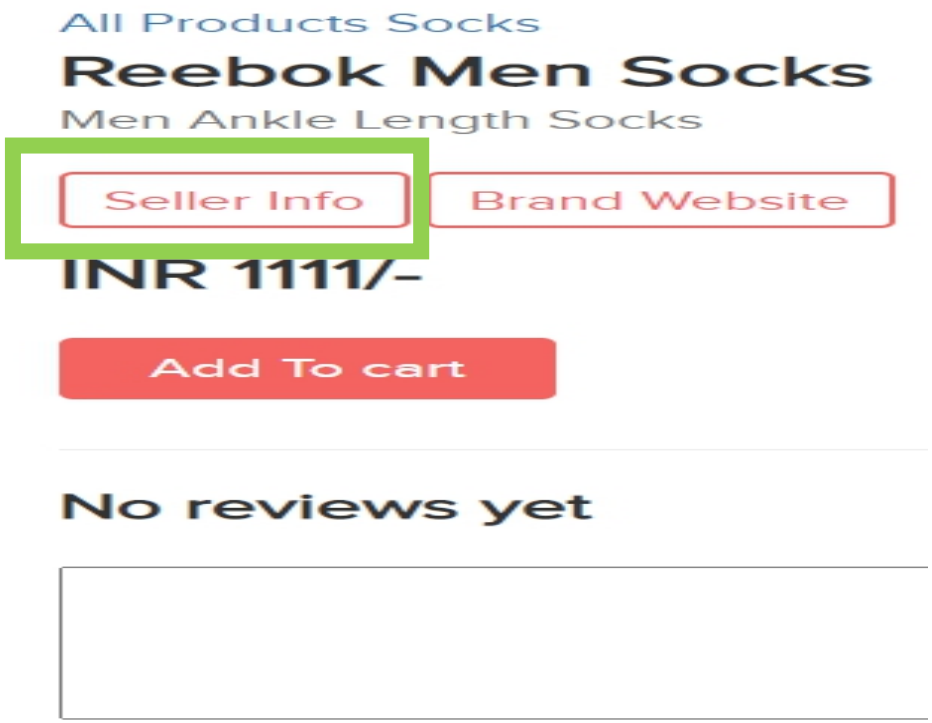
- https://www.acunetix.com/blog/articles/directory-listing-information-disclosure/

15.Personnel identifiable information-leakage

Personnel identifiable information	The url given below has PII-leakage • Affected url http://13.235.132.114/products/details.php?p_id=(here)

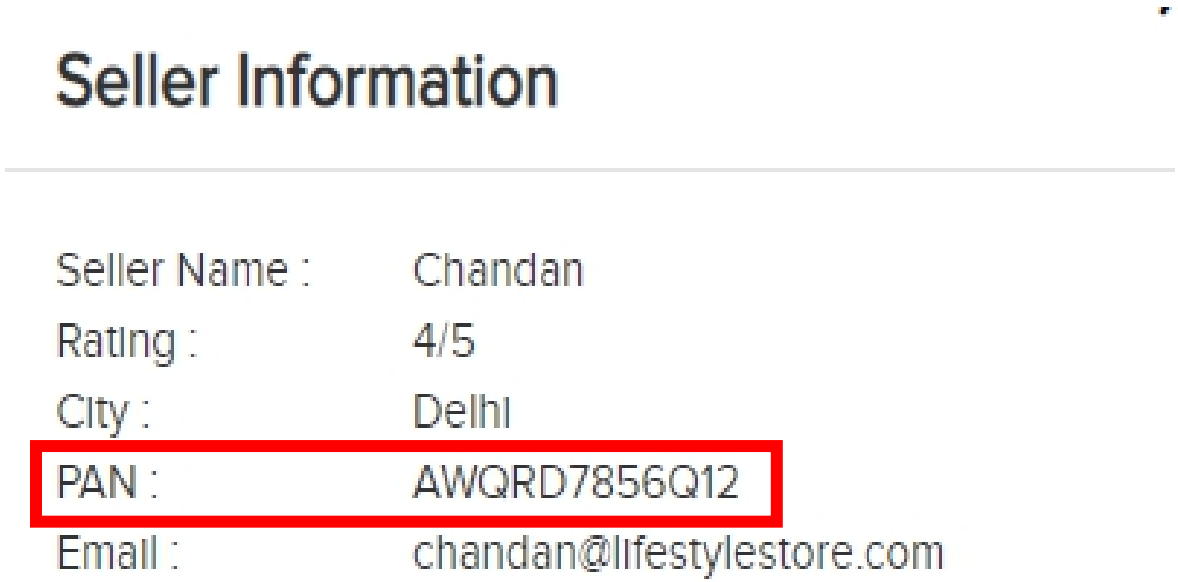
Observations

In every product pages the seller info option is available



POC

- Pan card details are also shown



Business impact - high

- Providing the Seller information may uninterest people to buy the item
- It may also cause social engineering attacks on seller

Recommendations

- Remove the pan card details
- Only show required information about anyone

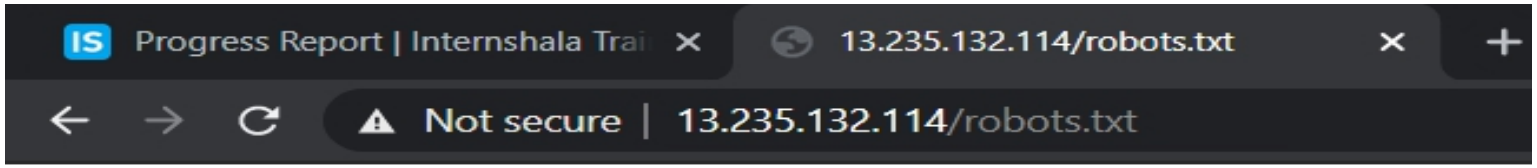
References

- https://www.imperva.com/learn/data-security/personally-identifiable-information-pii/
- https://hackerone.com/reports/374007

16.Default and debug files

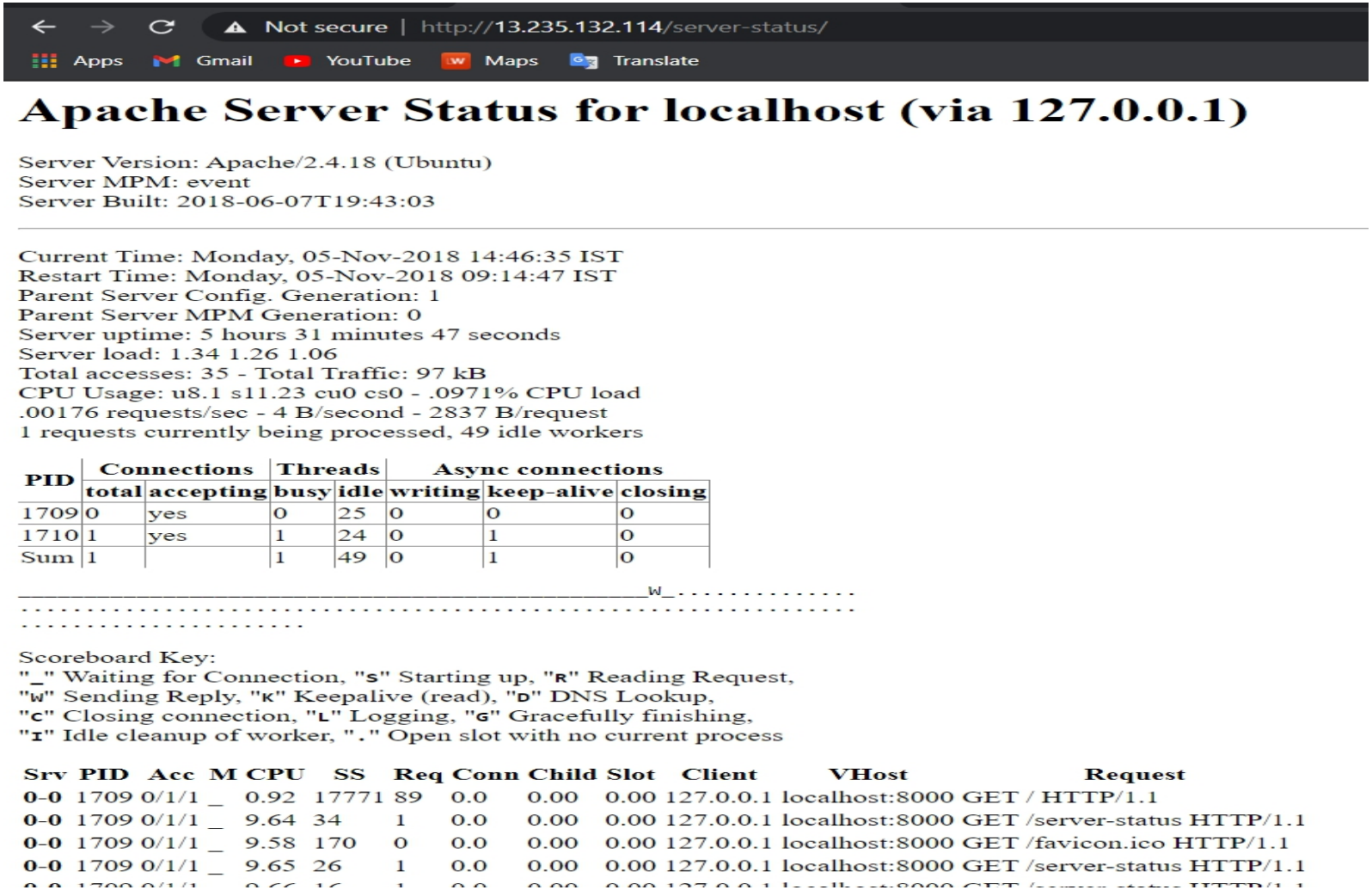
Default and debug files	Below mentione url has many default and debug files • Affected url http://13.235.132.114/ • Default pages 1. robots.txt 2. server-status 3. phpinfo.php 4. composer.json 5. userlist.txt

POC




```
User-Agent: *
Disallow: /static/images/
Disallow: /ovidentiaCMS
```

POC



Apache Server Status for localhost (via 127.0.0.1)

Server Version: Apache/2.4.18 (Ubuntu)
Server MPM: event
Server Built: 2018-06-07T19:43:03

Current Time: Monday, 05-Nov-2018 14:46:35 IST
Restart Time: Monday, 05-Nov-2018 09:14:47 IST
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 5 hours 31 minutes 47 seconds
Server load: 1.34 1.26 1.06
Total accesses: 35 - Total Traffic: 97 kB
CPU Usage: u8.1 s11.23 cu0 cs0 - .0971% CPU load
.00176 requests/sec - 4 B/second - 2837 B/request
1 requests currently being processed, 49 idle workers

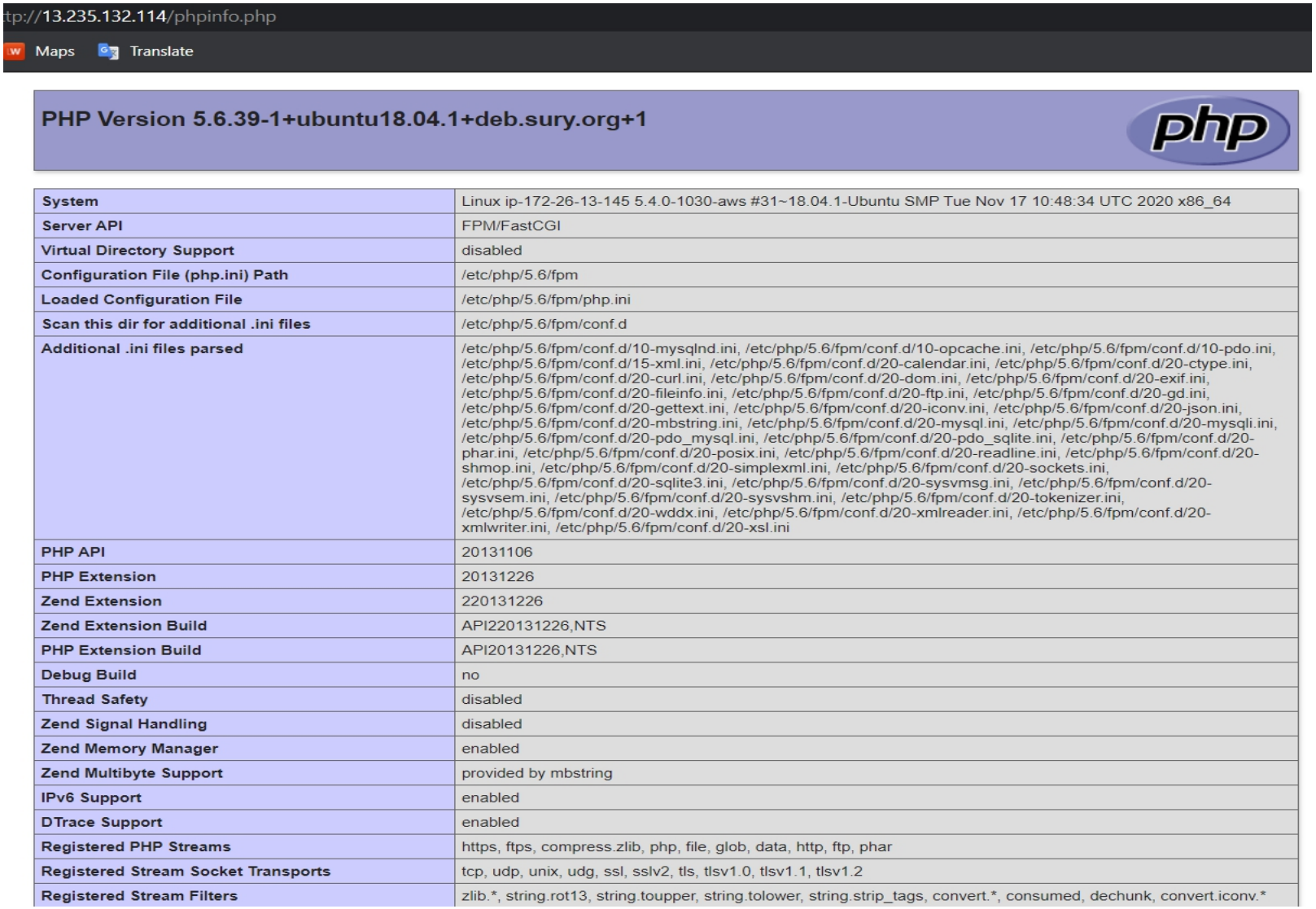
PID	Connections		Threads		Async connections		
	total	accepting	busy	idle	writing	keep-alive	closing
1709	0	yes	0	25	0	0	0
1710	1	yes	1	24	0	1	0
Sum	1		1	49	0	1	0

```
_____________________________________________________W_.............
................................................................................
.........................
```

Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

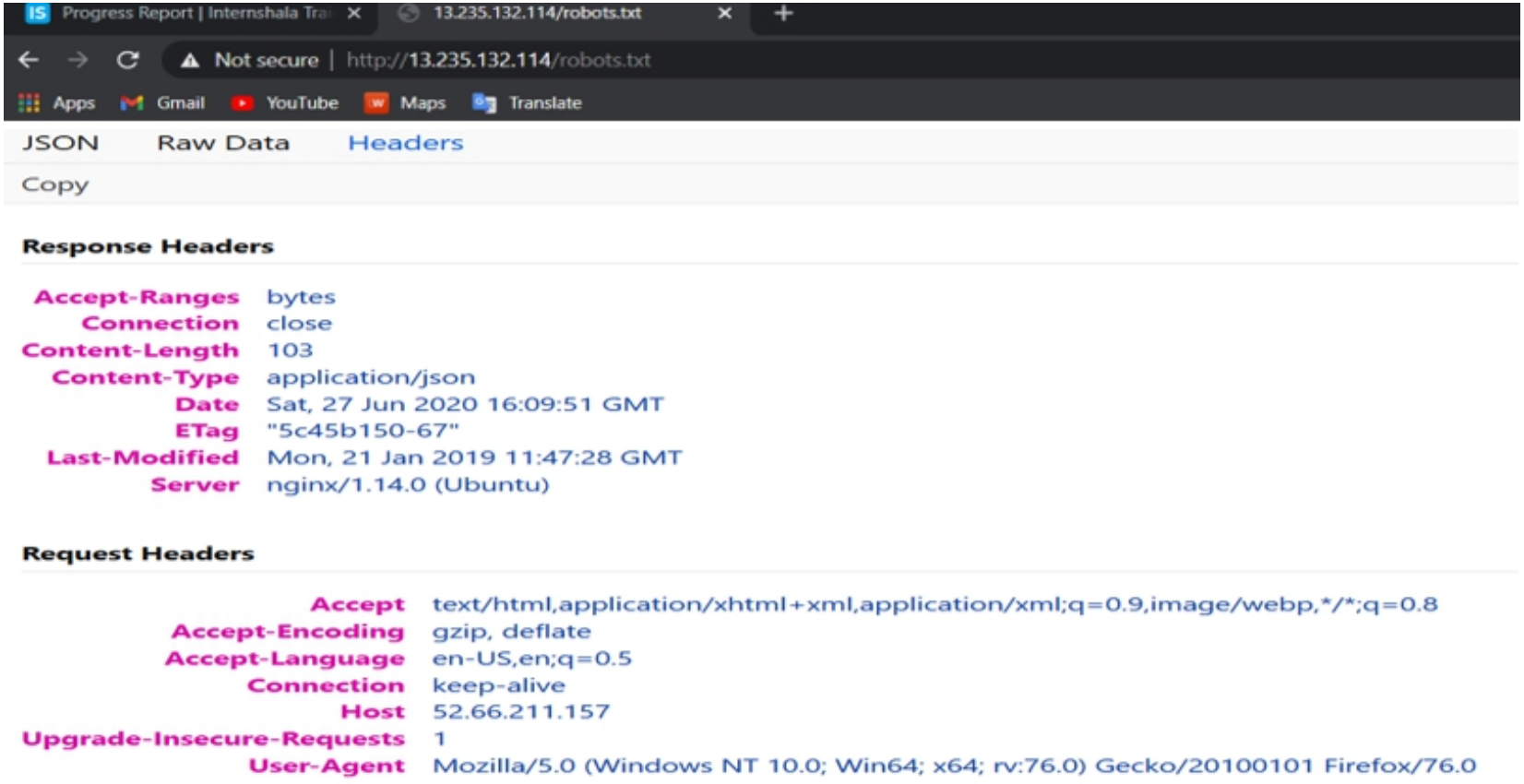
Srv	PID	Acc	M	CPU	SS	Req	Conn	Child	Slot	Client	VHost	Request
0-0	1709	0/1/1	_	0.92	17771	89	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET / HTTP/1.1
0-0	1709	0/1/1	_	9.64	34	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1
0-0	1709	0/1/1	_	9.58	170	0	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /favicon.ico HTTP/1.1
0-0	1709	0/1/1	_	9.65	26	1	0.0	0.00	0.00	127.0.0.1	localhost:8000	GET /server-status HTTP/1.1

POC

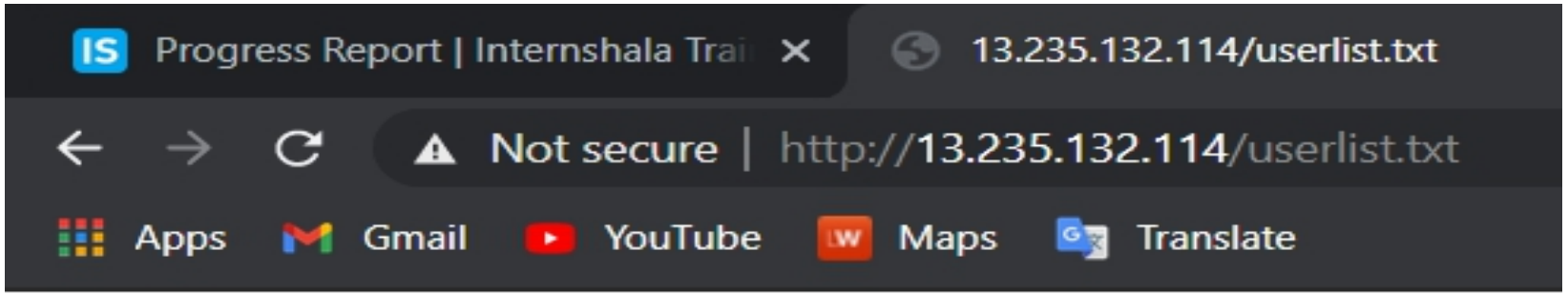


System	Linux ip-172-26-13-145 5.4.0-1030-aws #31~18.04.1-Ubuntu SMP Tue Nov 17 10:48:34 UTC 2020 x86_64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/5.6/fpm
Loaded Configuration File	/etc/php/5.6/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/5.6/fpm/conf.d
Additional .ini files parsed	/etc/php/5.6/fpm/conf.d/10-mysqlnd.ini, /etc/php/5.6/fpm/conf.d/10-opcache.ini, /etc/php/5.6/fpm/conf.d/10-pdo.ini, /etc/php/5.6/fpm/conf.d/15-xml.ini, /etc/php/5.6/fpm/conf.d/20-calendar.ini, /etc/php/5.6/fpm/conf.d/20-ctype.ini, /etc/php/5.6/fpm/conf.d/20-curl.ini, /etc/php/5.6/fpm/conf.d/20-dom.ini, /etc/php/5.6/fpm/conf.d/20-exif.ini, /etc/php/5.6/fpm/conf.d/20-fileinfo.ini, /etc/php/5.6/fpm/conf.d/20-ftp.ini, /etc/php/5.6/fpm/conf.d/20-gd.ini, /etc/php/5.6/fpm/conf.d/20-gettext.ini, /etc/php/5.6/fpm/conf.d/20-iconv.ini, /etc/php/5.6/fpm/conf.d/20-json.ini, /etc/php/5.6/fpm/conf.d/20-mbstring.ini, /etc/php/5.6/fpm/conf.d/20-mysql.ini, /etc/php/5.6/fpm/conf.d/20-mysqli.ini, /etc/php/5.6/fpm/conf.d/20-pdo_mysql.ini, /etc/php/5.6/fpm/conf.d/20-pdo_sqlite.ini, /etc/php/5.6/fpm/conf.d/20-phar.ini, /etc/php/5.6/fpm/conf.d/20-posix.ini, /etc/php/5.6/fpm/conf.d/20-readline.ini, /etc/php/5.6/fpm/conf.d/20-shmop.ini, /etc/php/5.6/fpm/conf.d/20-simplexml.ini, /etc/php/5.6/fpm/conf.d/20-sockets.ini, /etc/php/5.6/fpm/conf.d/20-sqlite3.ini, /etc/php/5.6/fpm/conf.d/20-sysvmsg.ini, /etc/php/5.6/fpm/conf.d/20-sysvsem.ini, /etc/php/5.6/fpm/conf.d/20-sysvshm.ini, /etc/php/5.6/fpm/conf.d/20-tokenizer.ini, /etc/php/5.6/fpm/conf.d/20-wddx.ini, /etc/php/5.6/fpm/conf.d/20-xmlreader.ini, /etc/php/5.6/fpm/conf.d/20-xmlwriter.ini, /etc/php/5.6/fpm/conf.d/20-xsl.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv2, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

POC


Progress Report | Internshala Tra
13.235.132.114/robots.txt
Not secure | http://13.235.132.114/robots.txt
Apps
Gmail
YouTube
Maps
Translate
JSON
Raw Data
Headers
Copy
Response Headers
Accept-Ranges bytes
Connection close
Content-Length 103
Content-Type application/json
Date Sat, 27 Jun 2020 16:09:51 GMT
ETag "5c45b150-67"
Last-Modified Mon, 21 Jan 2019 11:47:28 GMT
Server nginx/1.14.0 (Ubuntu)
Request Headers
Accept text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding gzip, deflate
Accept-Language en-US,en;q=0.5
Connection keep-alive
Host 52.66.211.157
Upgrade-Insecure-Requests 1
User-Agent Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:76.0) Gecko/20100101 Firefox/76.0

POC




```
Radhika:Radhika123:6
Nandan:Nandan123:7
chandan:chandan123:4
```

Business impact - low

- It does not impact the website directly
- It only helps hacker to collect information

Recommendations

- Disable all these default pages

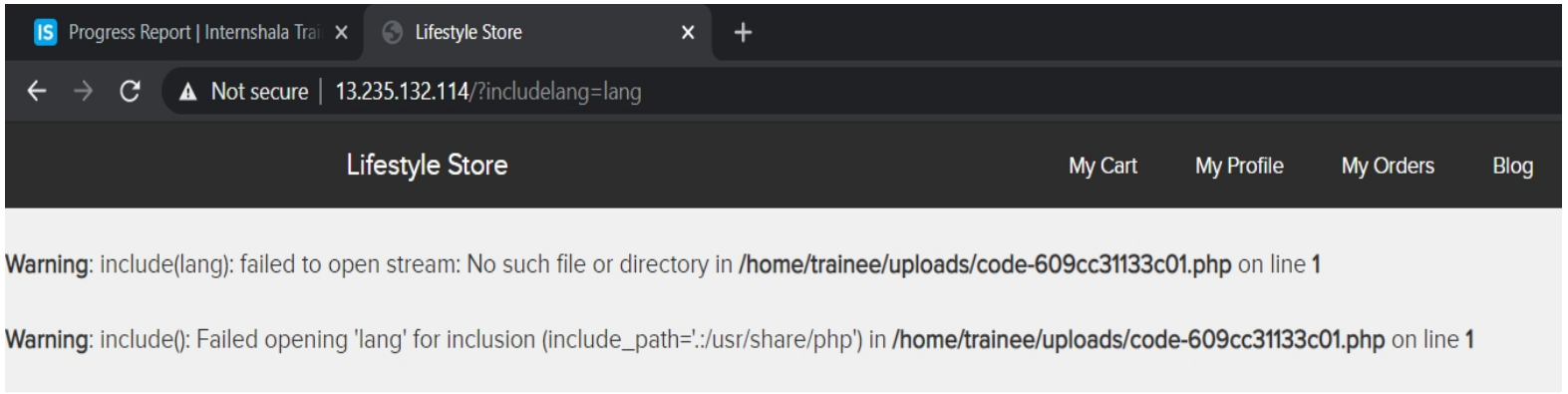
References

- https://www.indusface.com/blog/owasp-security-misconfiguration/
- https://hdivsecurity.com/owasp-security-misconfiguration

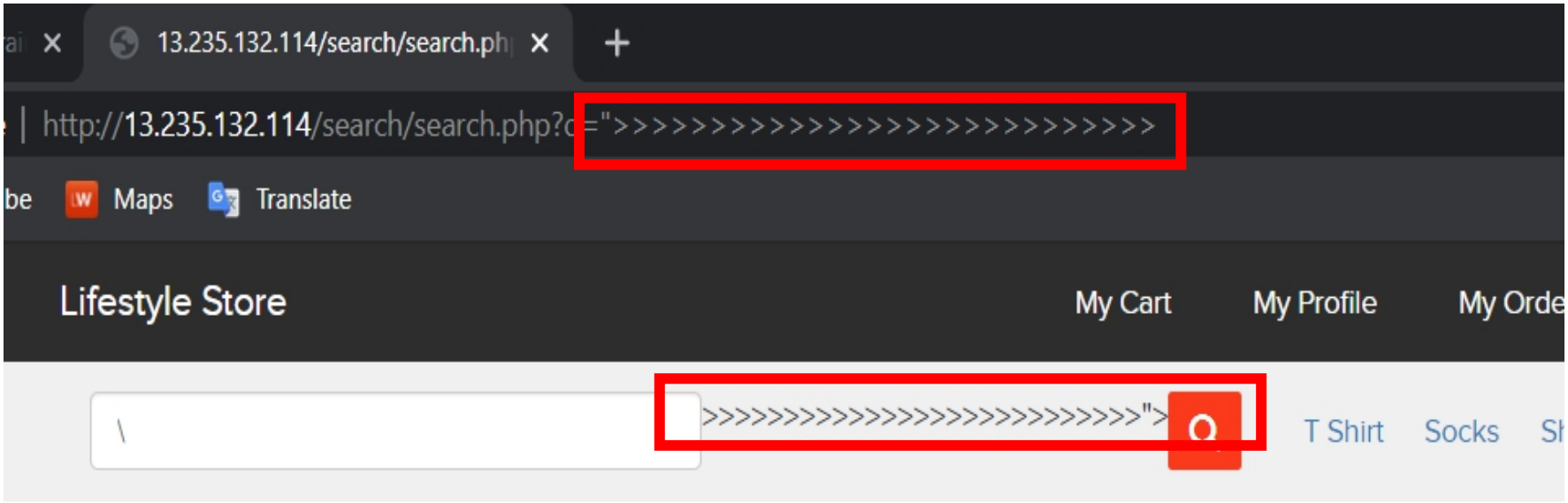
17.Descriptive error messages

Descriptive error messages	Below mentioned url shows Descriptive error messages • Affected url http://13.235.132.114/?includelang=lang http://13.235.132.114/search/search.php?q=(here)

POC


Progress Report | Internshala Trai
Lifestyle Store
Not secure | 13.235.132.114/?includelang=lang
Lifestyle Store
My Cart
My Profile
My Orders
Blog
Warning: include(lang): failed to open stream: No such file or directory in /home/trainee/uploads/code-609cc31133c01.php on line 1
Warning: include(): Failed opening 'lang' for inclusion (include_path='.:/usr/share/php') in /home/trainee/uploads/code-609cc31133c01.php on line 1

POC


13.235.132.114/search/search.ph
http://13.235.132.114/search/search.php?c=">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
Maps
Translate
Lifestyle Store
My Cart
My Profile
My Orde
\
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>">
T Shirt
Socks

Business impact-low

- Although this vulnerability does not have a direct impact to users or the server, though it can help the attacker in mapping the server architecture and plan further attacks on the server.
- It doesn’t harm the website directly
- But it is letting the hacker to know about the website architecture

Recommendations

- Block these kind of error pages to show up
- Only show simple error pages
- Do not display the default error messages because it not tells about the server but also sometimes about the location.So, whenever there is an error ,send it to the same page or throw some manually written error.

References

- https://owasp.org/www-community/Improper_Error_Handling#:~:text=Description,to%20the%20user%20(hacker).
- https://cwe.mitre.org/data/definitions/209.html

Thank you

For further classification / patch assistance
contact - 6353XXXXXX